TO620T-T695S60T

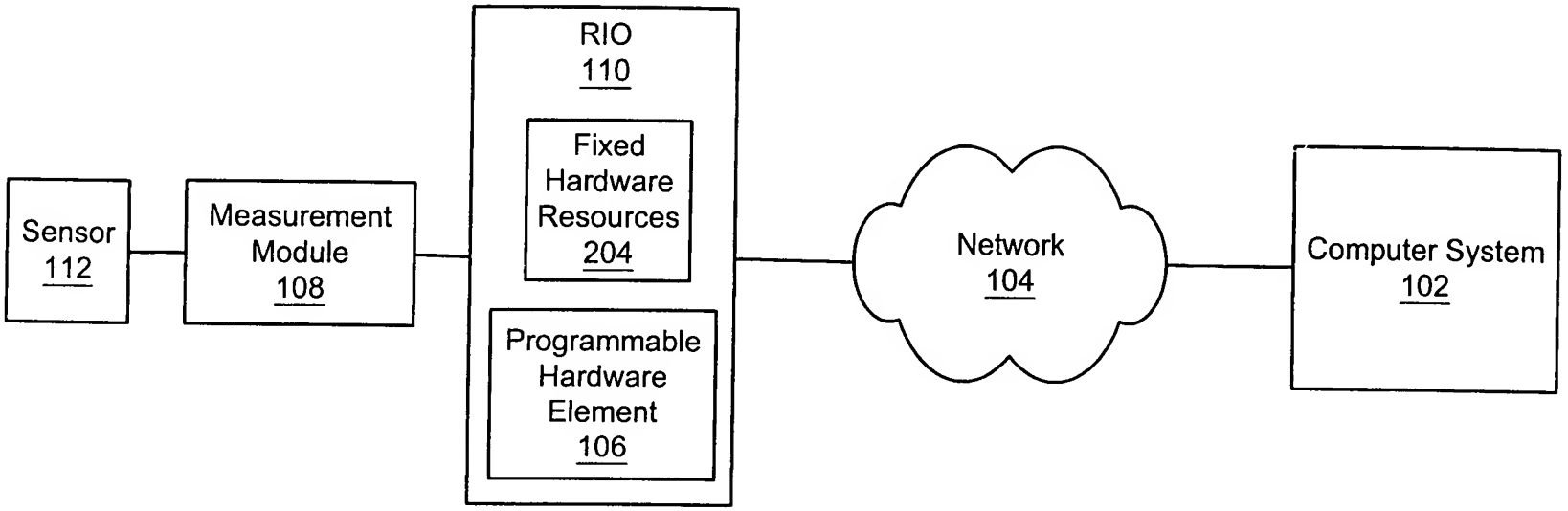

Figure 1

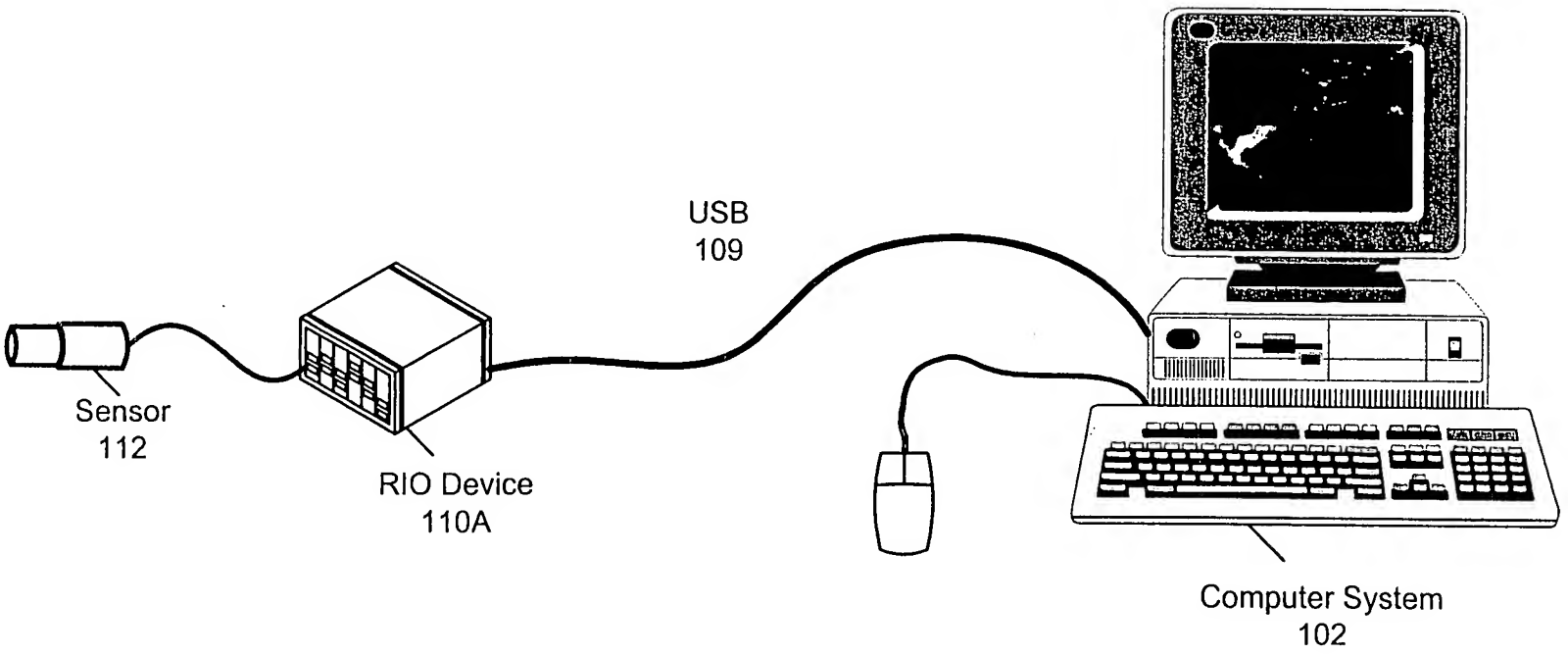

Figure 1A

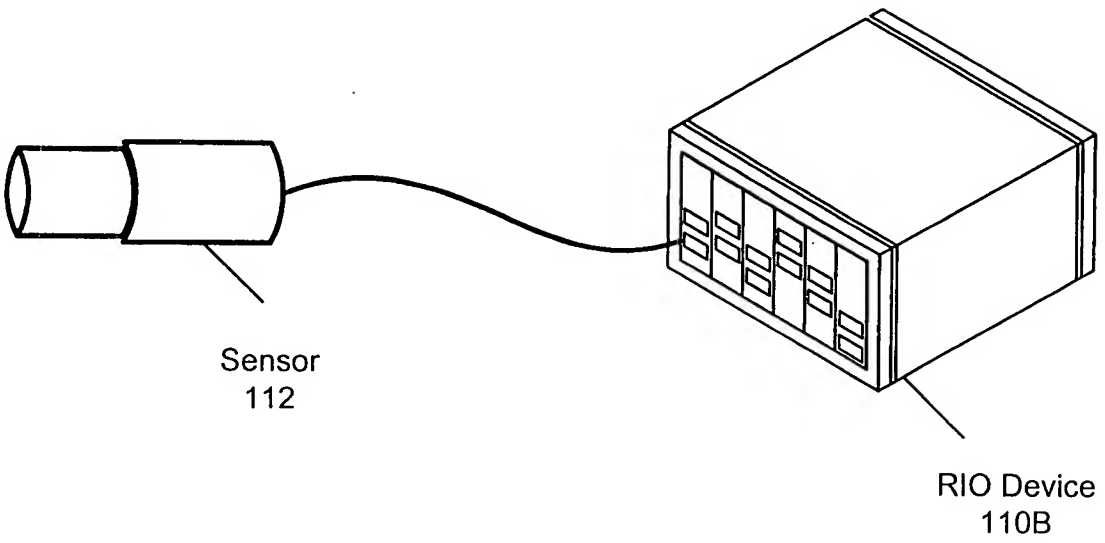

Figure 1B

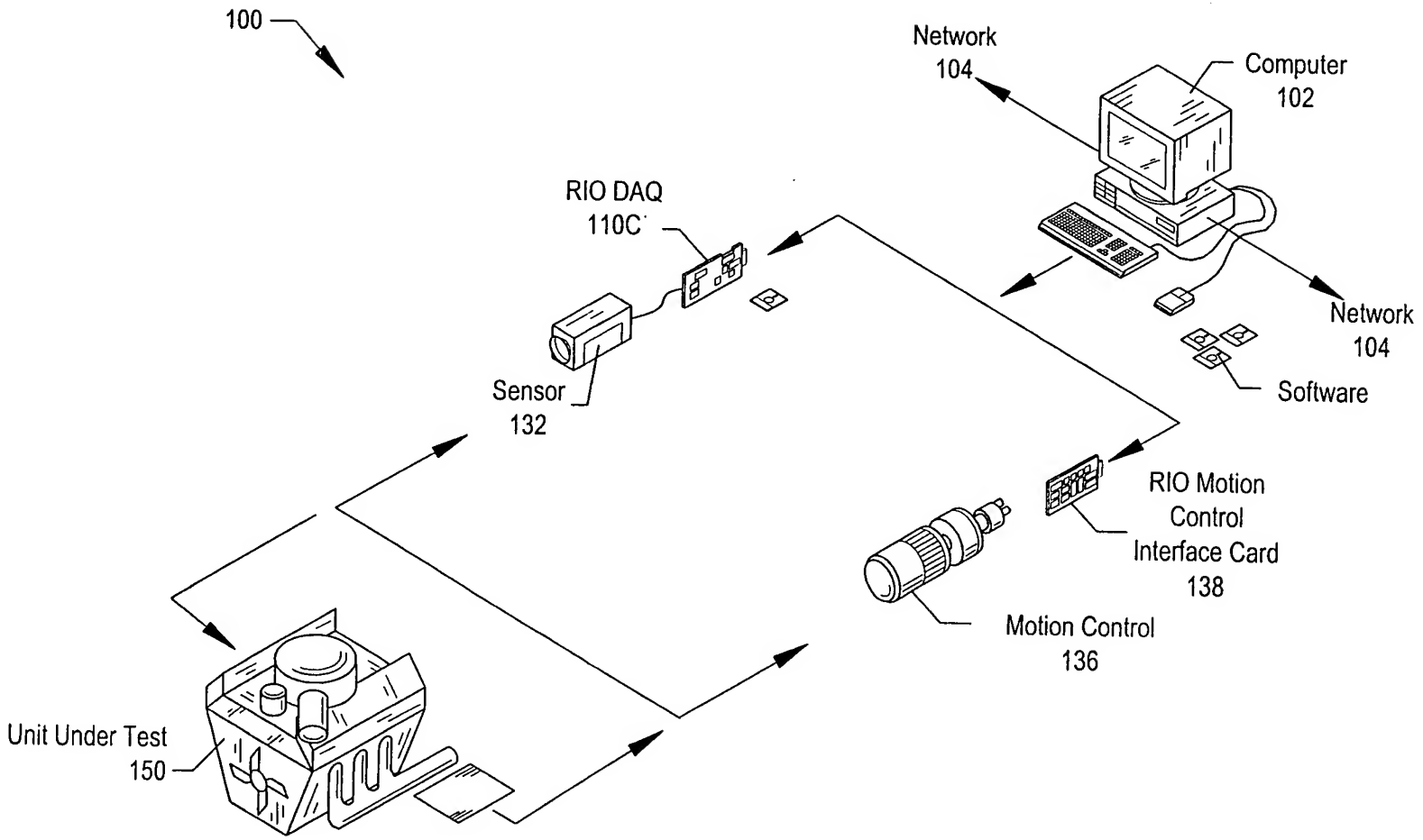
100
Network
104
Computer
102
RIO DAQ
110C
Sensor
132
Network
104
Software
RIO Motion
Control
Interface Card
138
Motion Control
136
Unit Under Test
150

Figure 1C

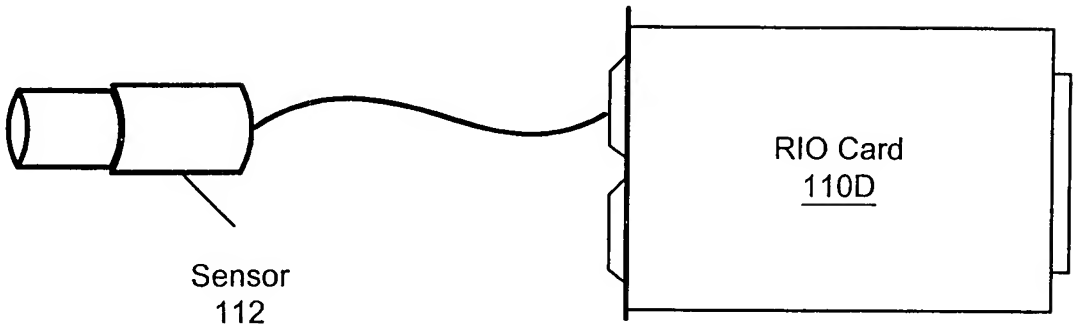

Figure 1D

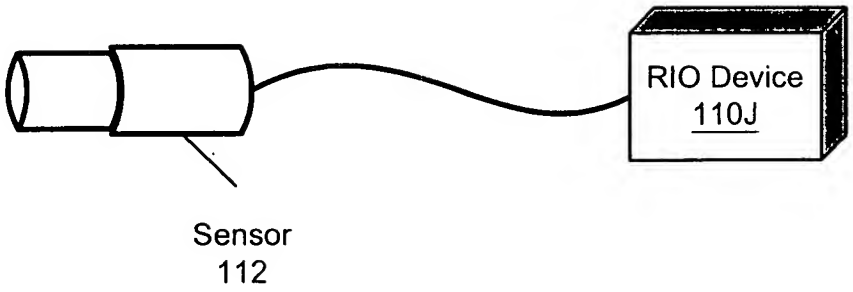

Figure 1E

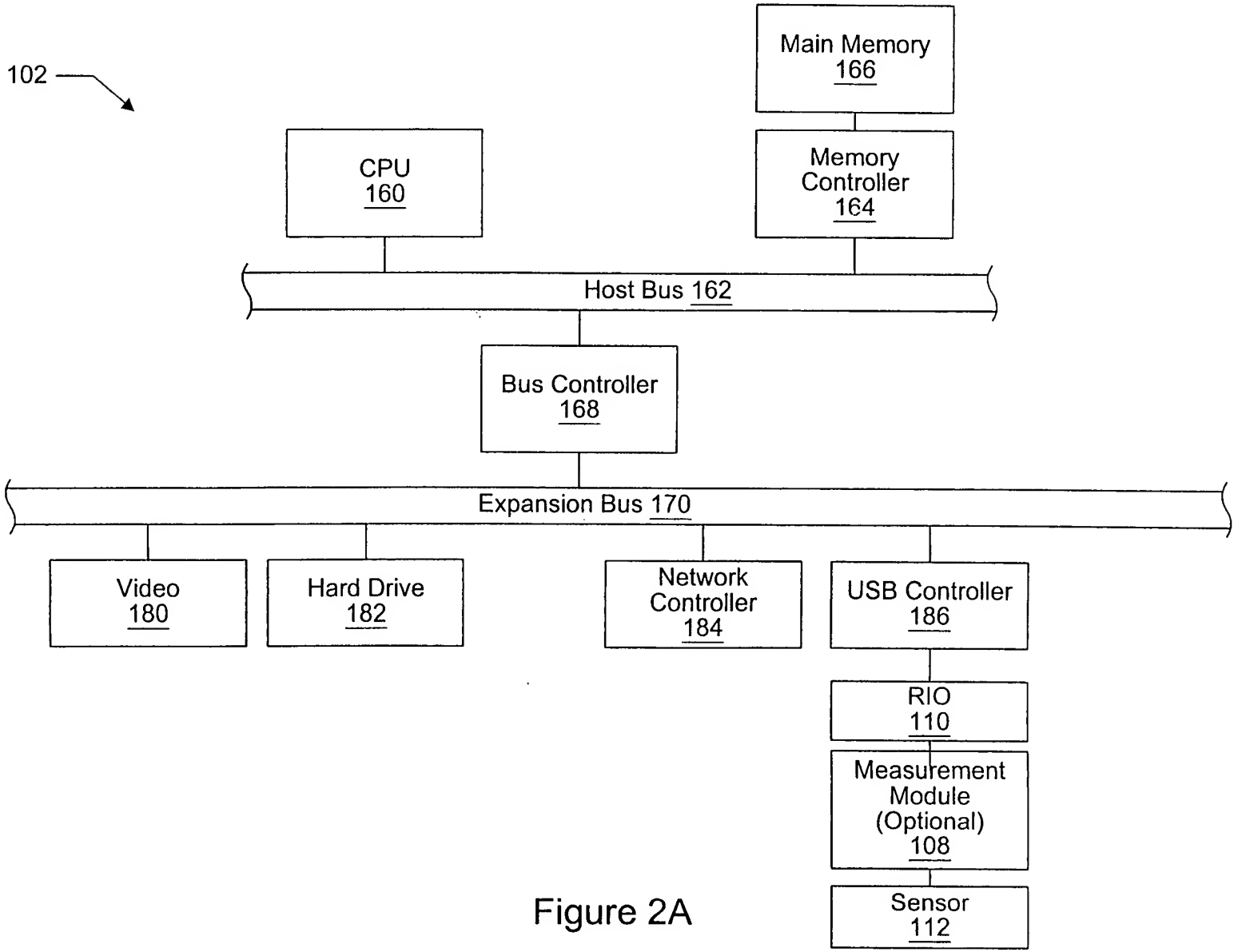
102
Main Memory
166
CPU
160
Memory
Controller
164
Host Bus 162
Bus Controller
168
Expansion Bus 170
Video
180
Hard Drive
182
Network
Controller
184
USB Controller
186
RIO
110
Measurement
Module
(Optional)
108
Sensor
112

Figure 2A

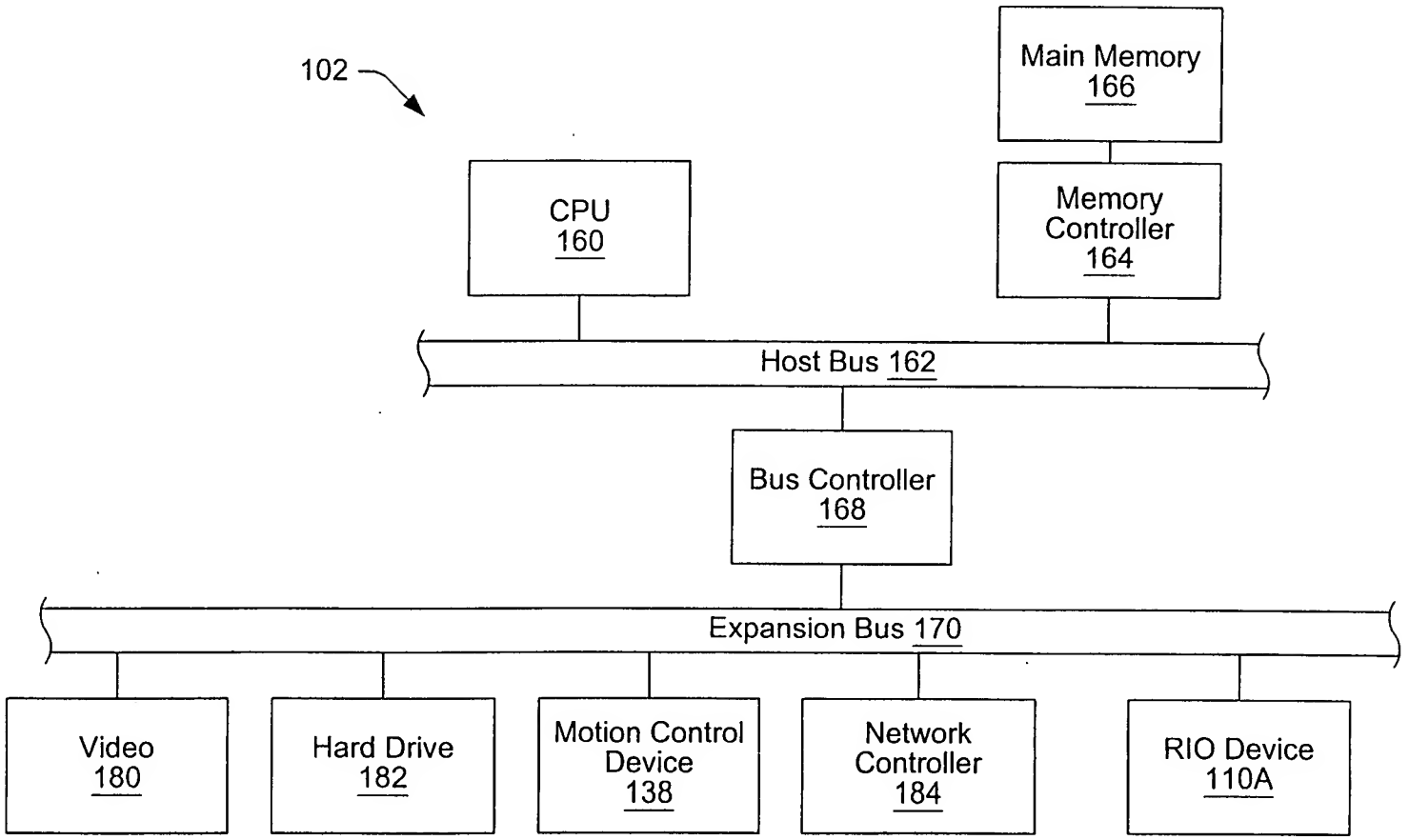
102
Main Memory
166
CPU
160
Memory
Controller
164
Host Bus 162
Bus Controller
168
Expansion Bus 170
Video
180
Hard Drive
182
Motion Control
Device
138
Network
Controller
184
RIO Device
110A

Figure 2B

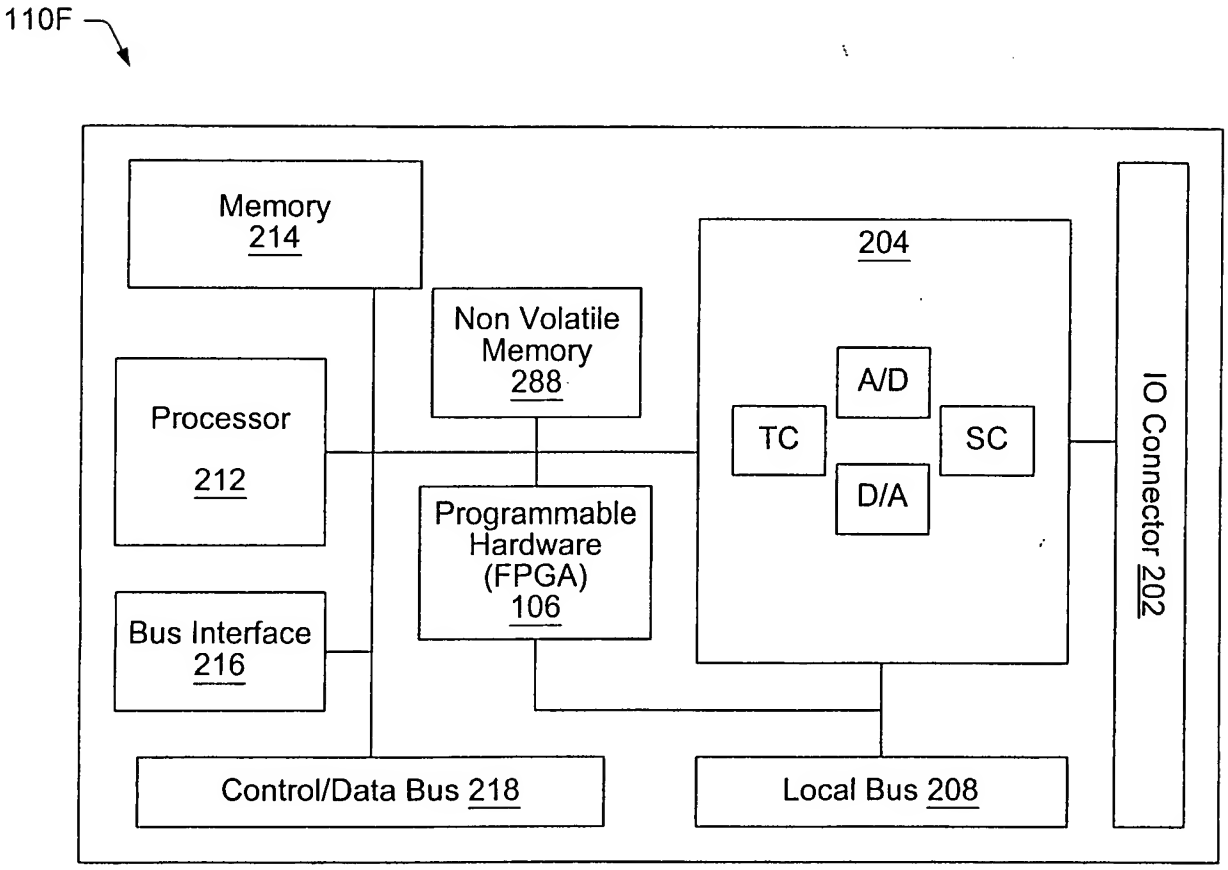
110F
Memory
214
Non Volatile
Memory
288
Processor
212
Programmable
Hardware
(FPGA)
106
Bus Interface
216
204
A/D
TC
SC
D/A
IO Connector 202
Control/Data Bus 218
Local Bus 208

Fig. 3B

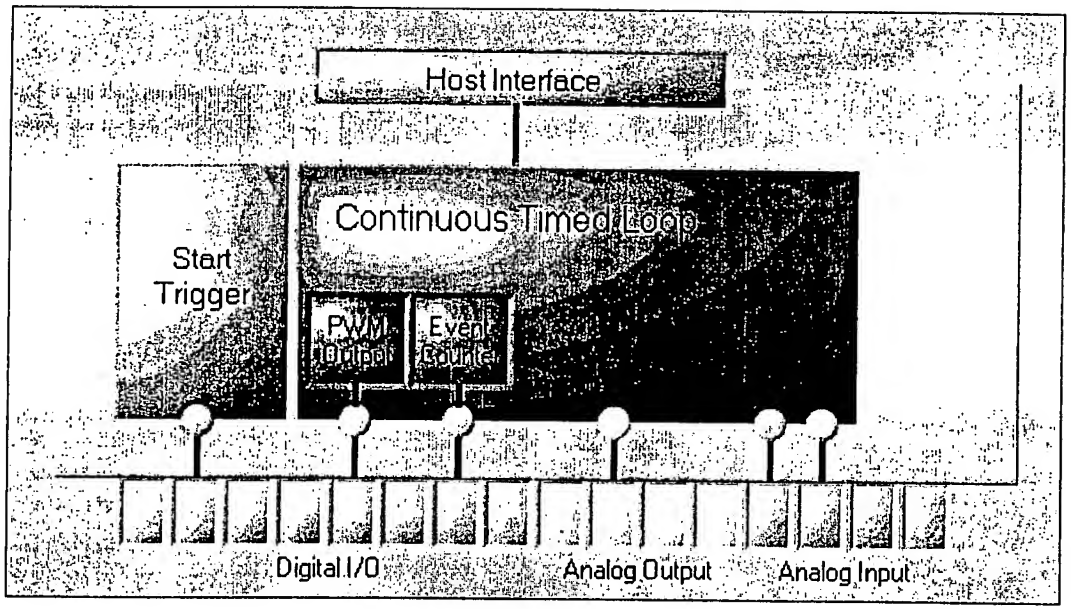
Host Interface
Continuous Timed Loop
Start Trigger
PWM Output
Event Counter
Digital I/O
Analog Output
Analog Input

Figure 3C

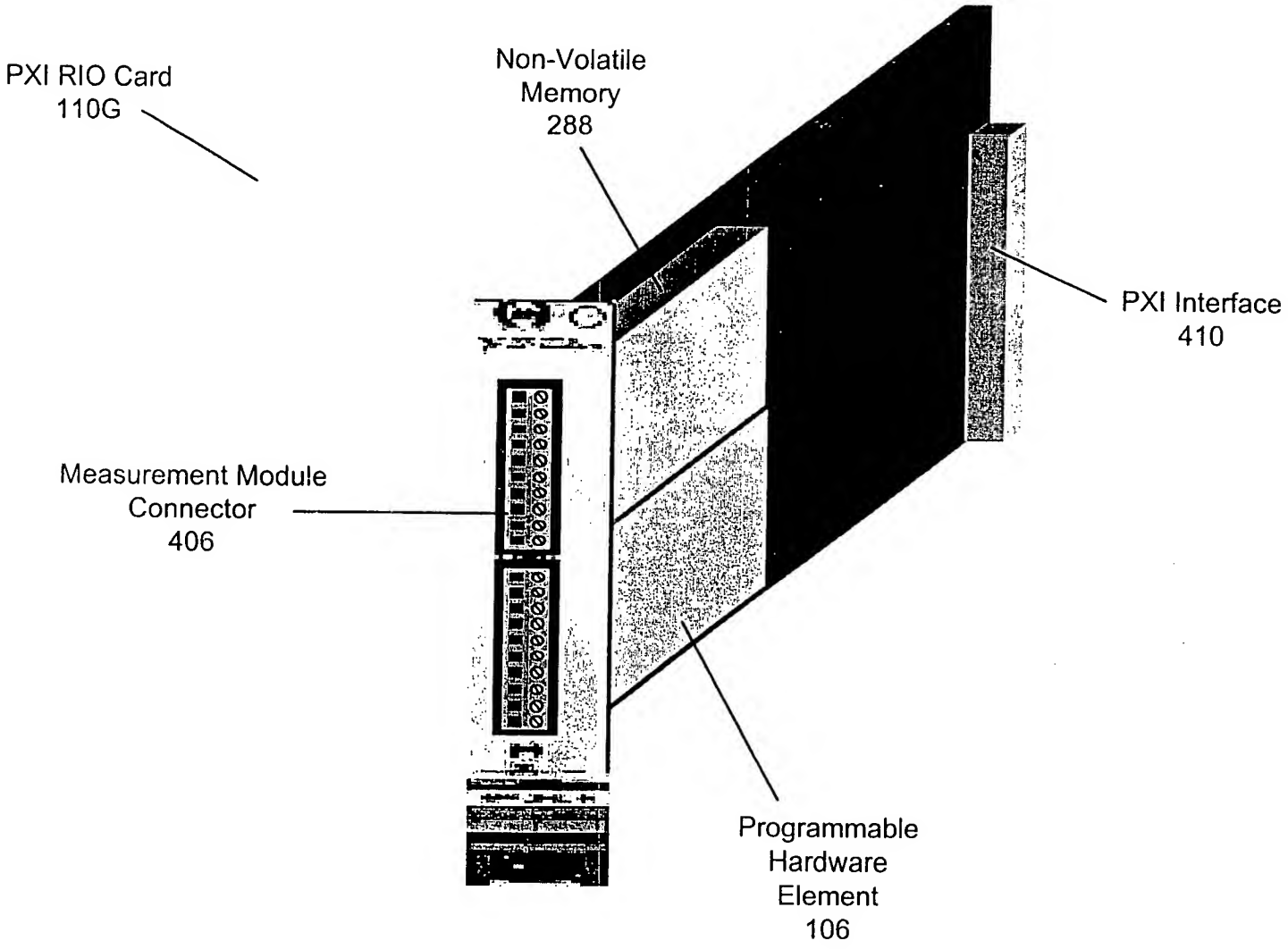
PXI RIO Card
110G
Non-Volatile
Memory
288
PXI Interface
410
Measurement Module
Connector
406
Programmable
Hardware
Element
106

Figure 4

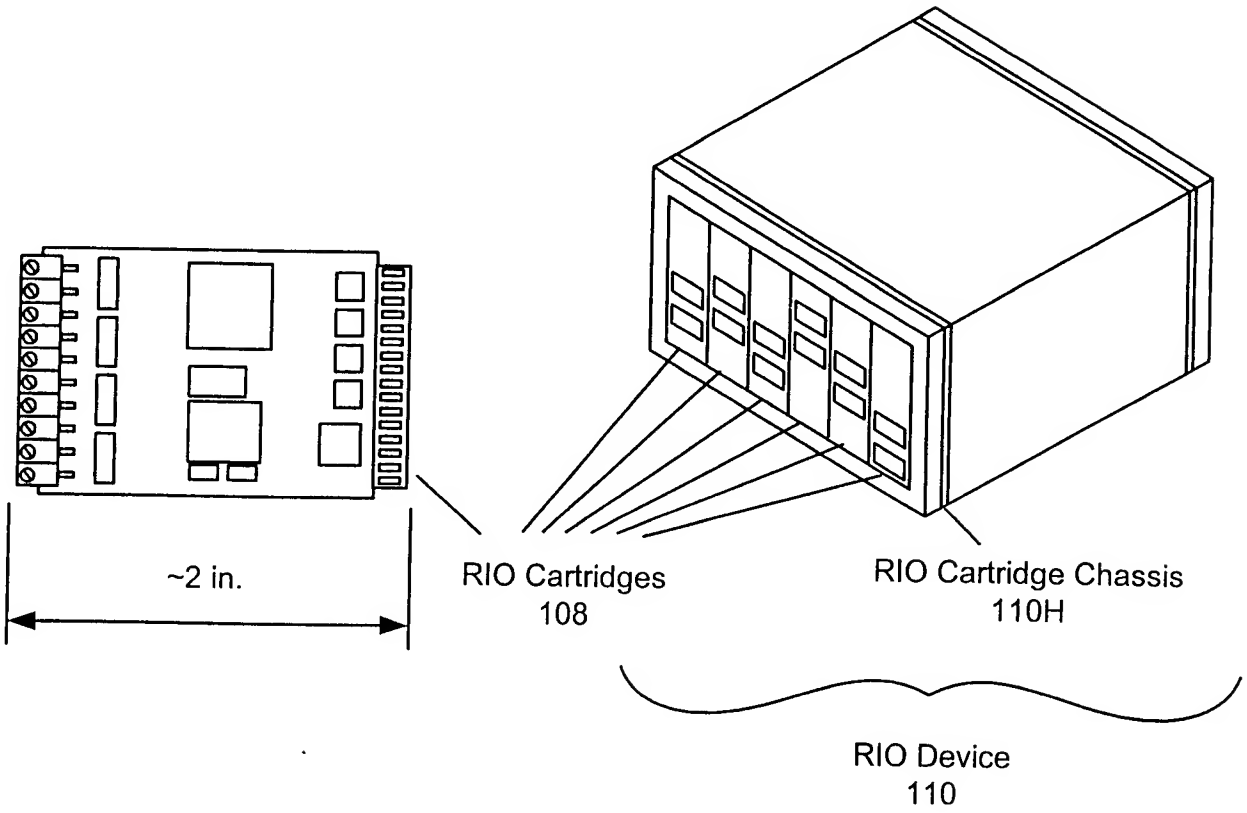
~2 in.
RIO Cartridges
108
RIO Cartridge Chassis
110H
RIO Device
110

Figure 5A

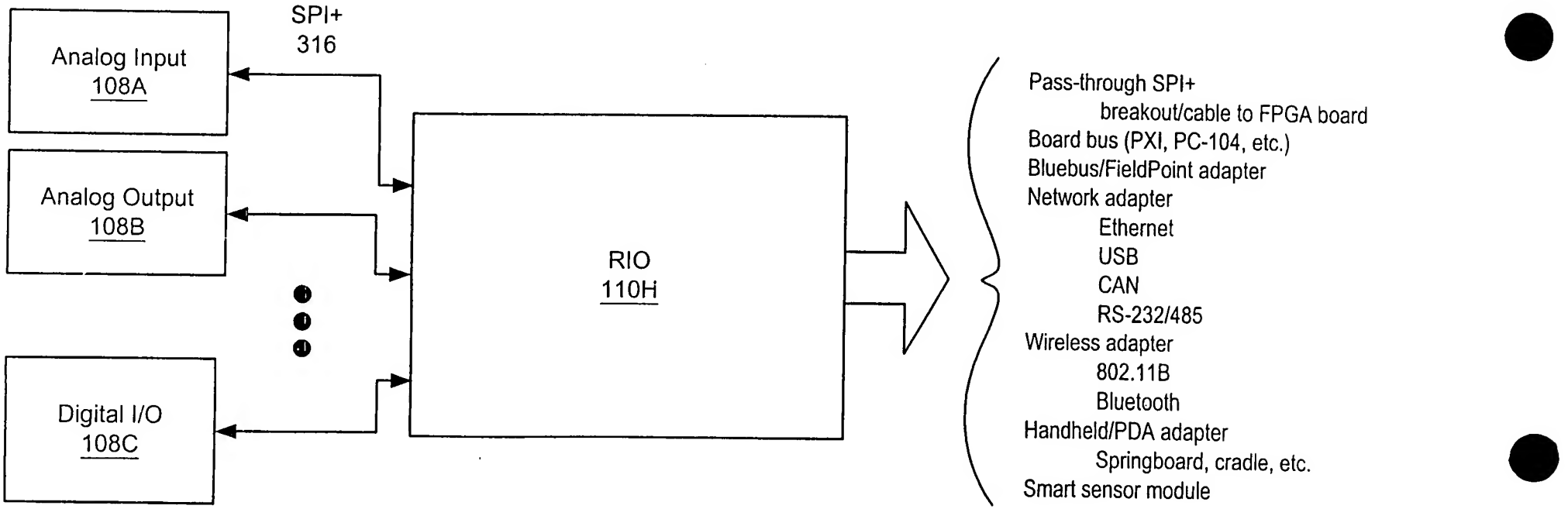
SPI+
316
Analog Input
108A
Analog Output
108B
Digital I/O
108C
RIO
110H
Pass-through SPI+
breakout/cable to FPGA board
Board bus (PXI, PC-104, etc.)
Bluebus/FieldPoint adapter
Network adapter
Ethernet
USB
CAN
RS-232/485
Wireless adapter
802.11B
Bluetooth
Handheld/PDA adapter
Springboard, cradle, etc.
Smart sensor module

Figure 5B

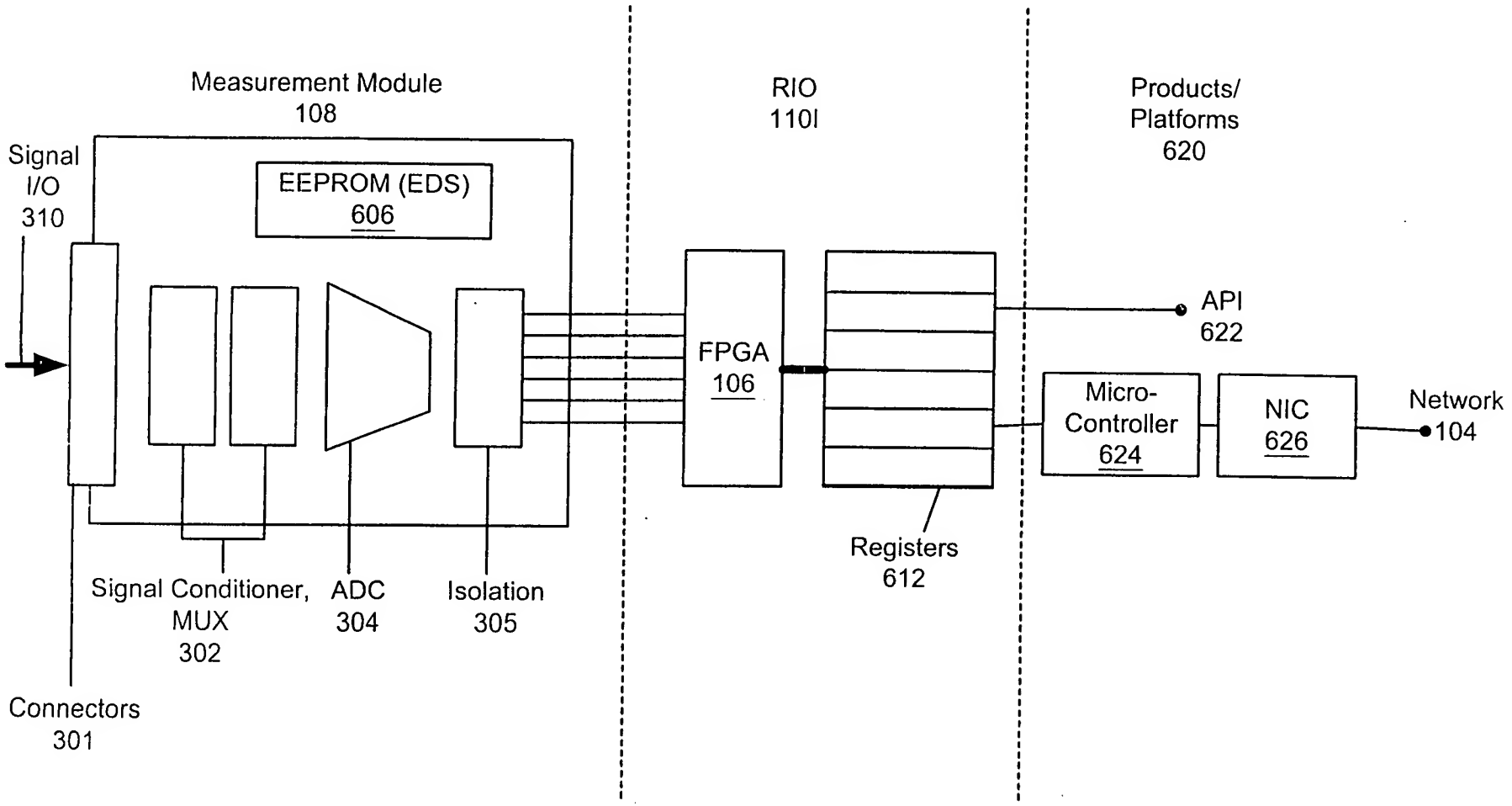
Measurement Module
108
RIO
110I
Products/
Platforms
620
Signal
I/O
310
EEPROM (EDS)
606
FPGA
106
API
622
Micro-
Controller
624
NIC
626
Network
104
Registers
612
Signal Conditioner,
MUX
302
ADC
304
Isolation
305
Connectors
301

Figure 6

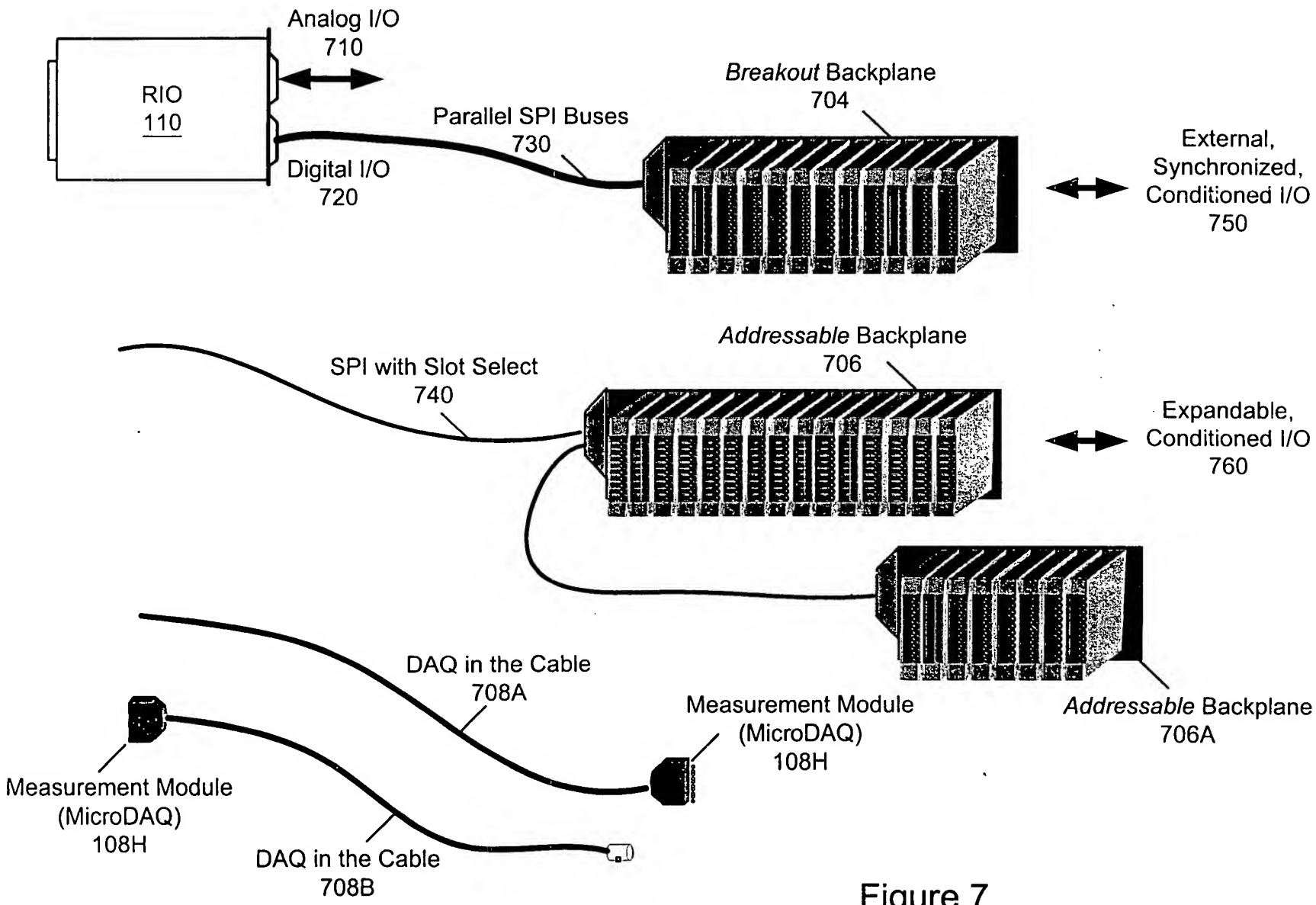
Analog I/O
710
RIO
110
Digital I/O
720
Parallel SPI Buses
730
Breakout Backplane
704
External,
Synchronized,
Conditioned I/O
750
SPI with Slot Select
740
Addressable Backplane
706
Expandable,
Conditioned I/O
760
DAQ in the Cable
708A
Measurement Module
(MicroDAQ)
108H
Addressable Backplane
706A
Measurement Module
(MicroDAQ)
108H
DAQ in the Cable
708B

Figure 7

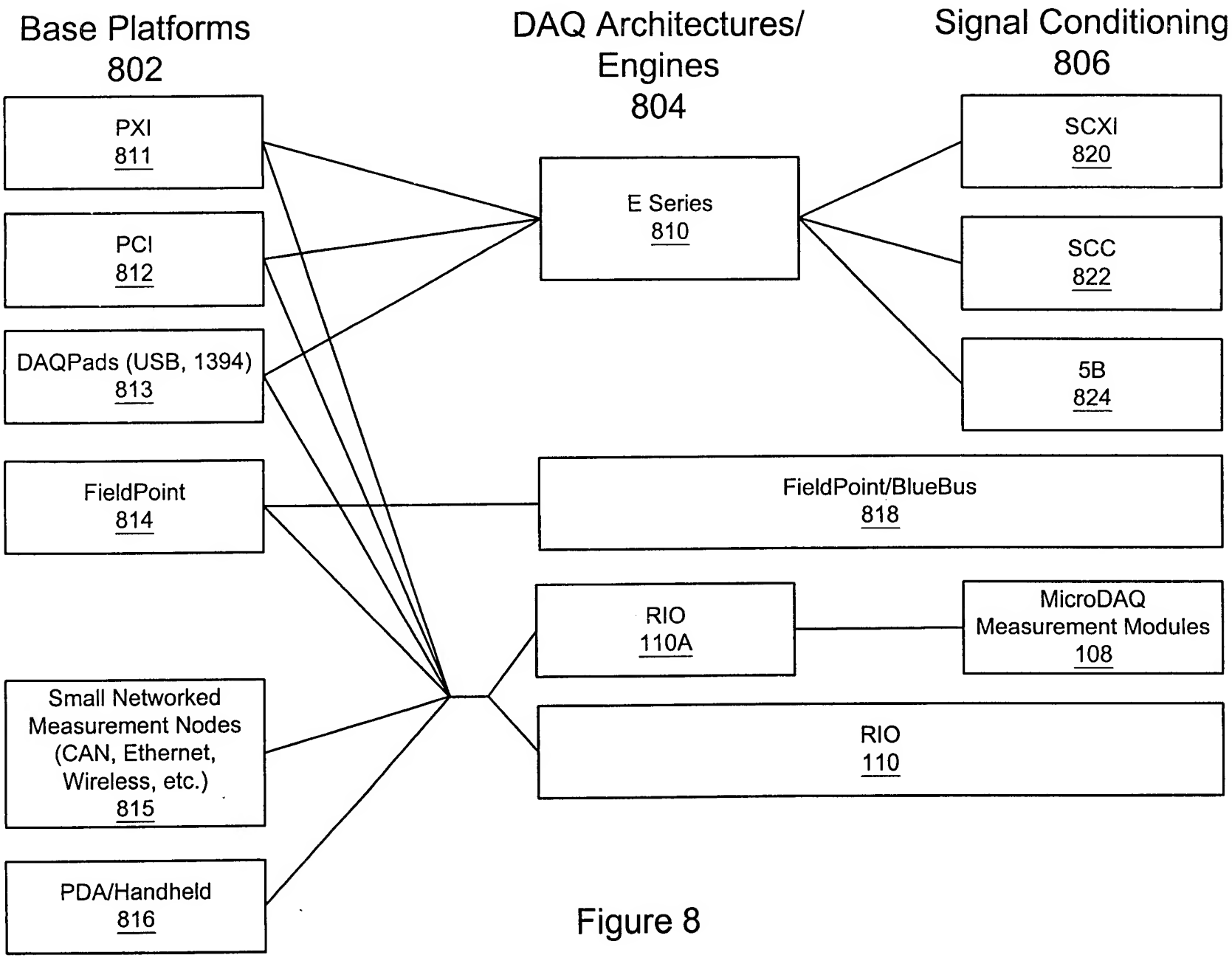
Base Platforms
802
PXI
811
PCI
812
DAQPads (USB, 1394)
813
FieldPoint
814
Small Networked
Measurement Nodes
(CAN, Ethernet,
Wireless, etc.)
815
PDA/Handheld
816
DAQ Architectures/
Engines
804
E Series
810
FieldPoint/BlueBus
818
RIO
110A
RIO
110
Signal Conditioning
806
SCXI
820
SCC
822
5B
824
MicroDAQ
Measurement Modules
108

Figure 8

4005654.1 10/26/04

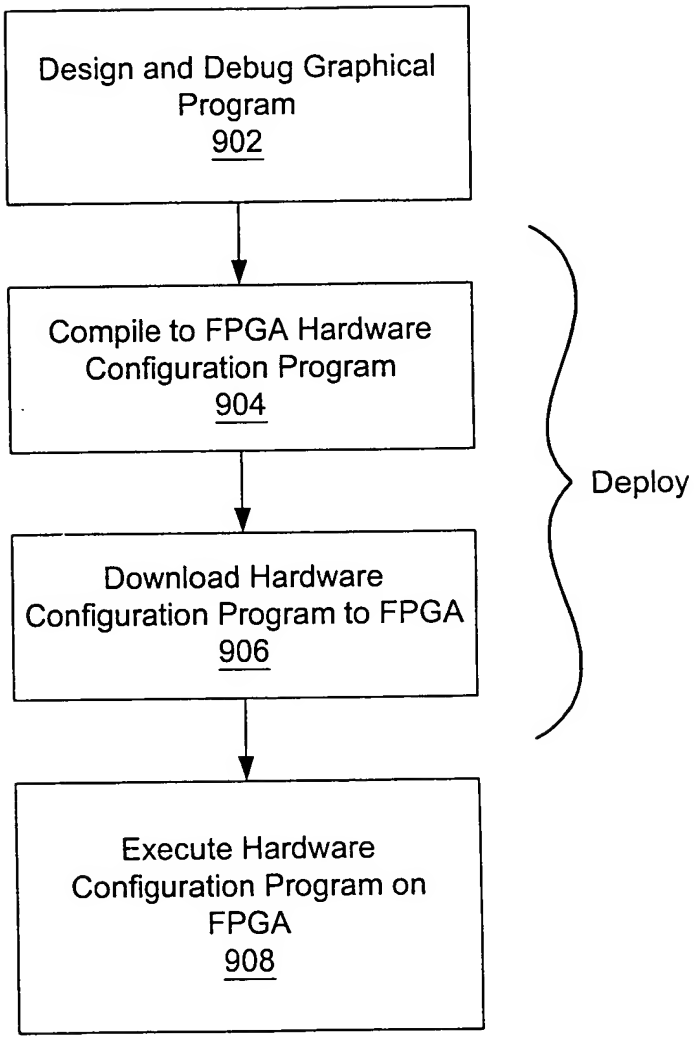

Figure 9

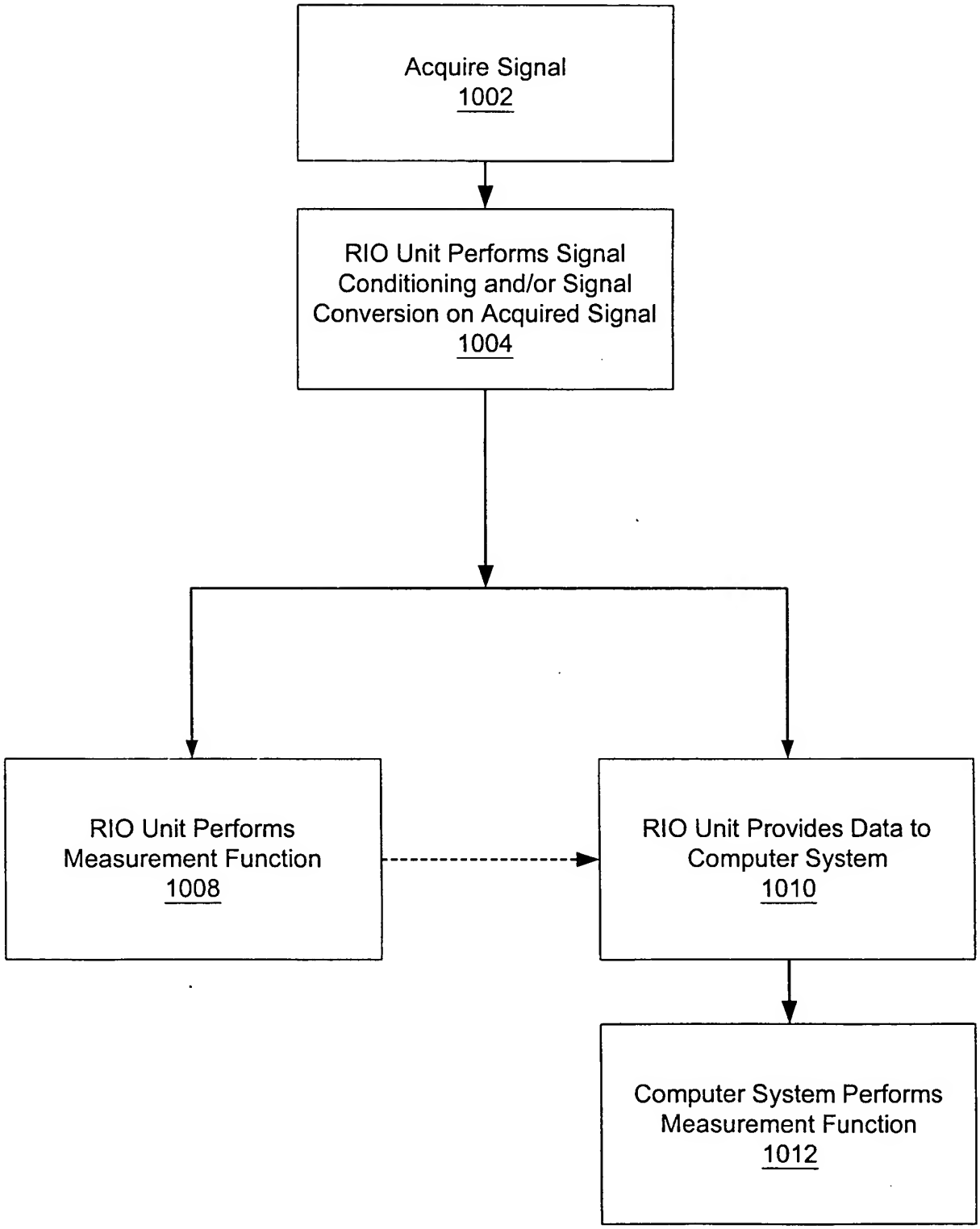
Acquire Signal
1002
RIO Unit Performs Signal Conditioning and/or Signal Conversion on Acquired Signal
1004
RIO Unit Performs Measurement Function
1008
RIO Unit Provides Data to Computer System
1010
Computer System Performs Measurement Function
1012

Figure 10

4005504 - 4020504

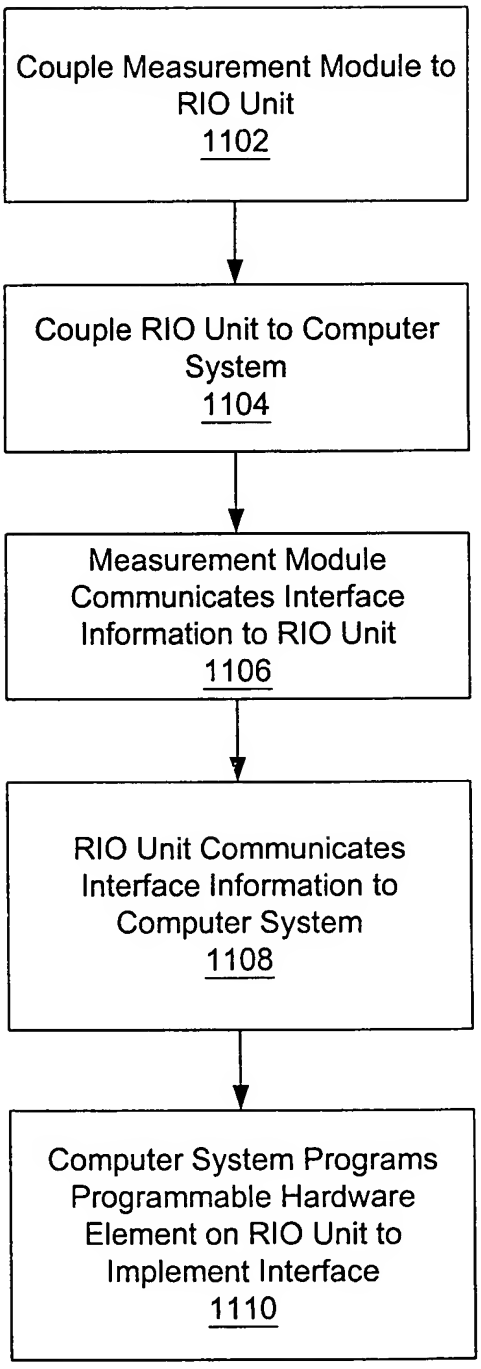

Figure 11A

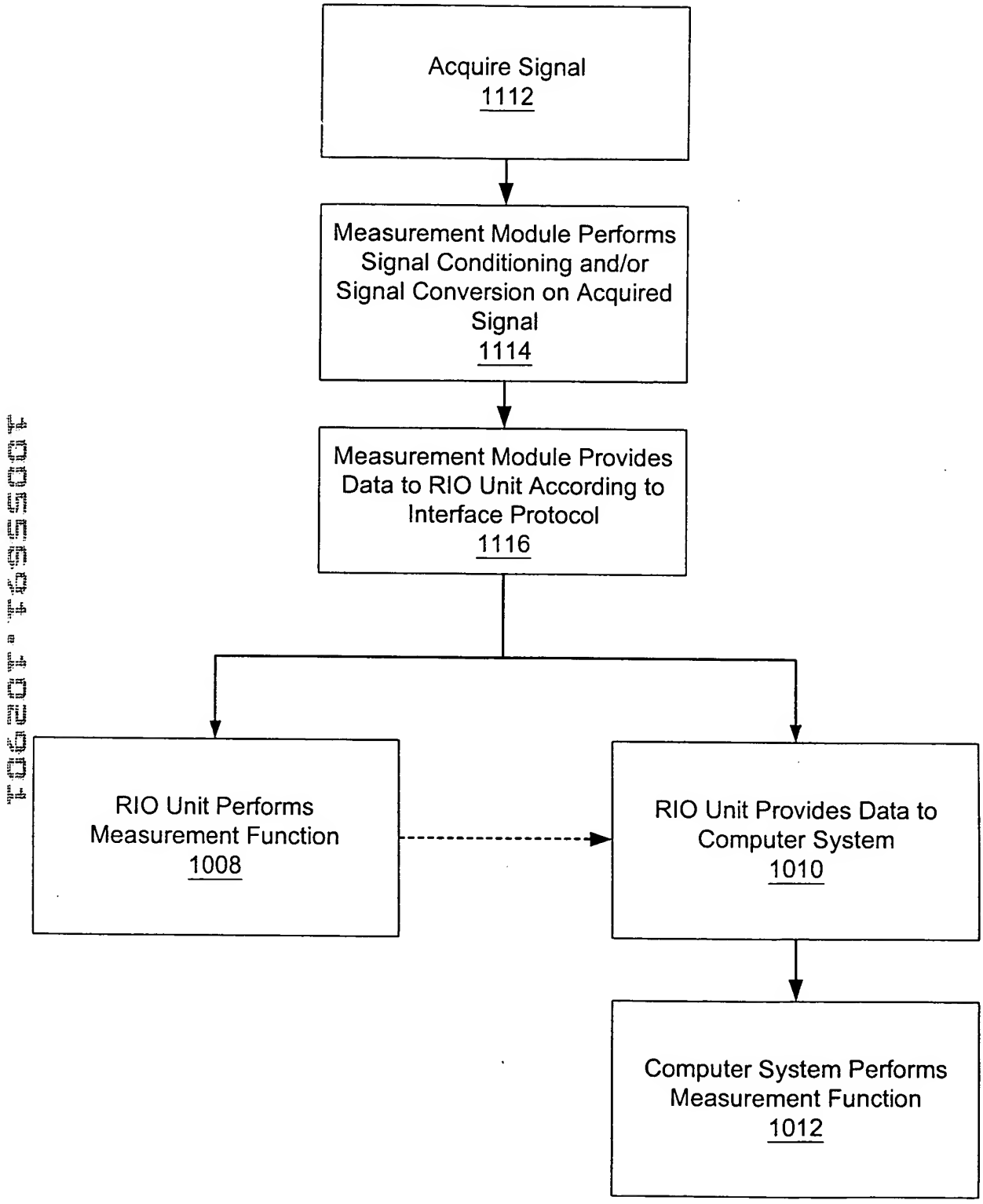
Acquire Signal
1112
Measurement Module Performs Signal Conditioning and/or Signal Conversion on Acquired Signal
1114
Measurement Module Provides Data to RIO Unit According to Interface Protocol
1116
RIO Unit Performs Measurement Function
1008
RIO Unit Provides Data to Computer System
1010
Computer System Performs Measurement Function
1012

Figure 11B

T0620T" T6955001

Figure 12

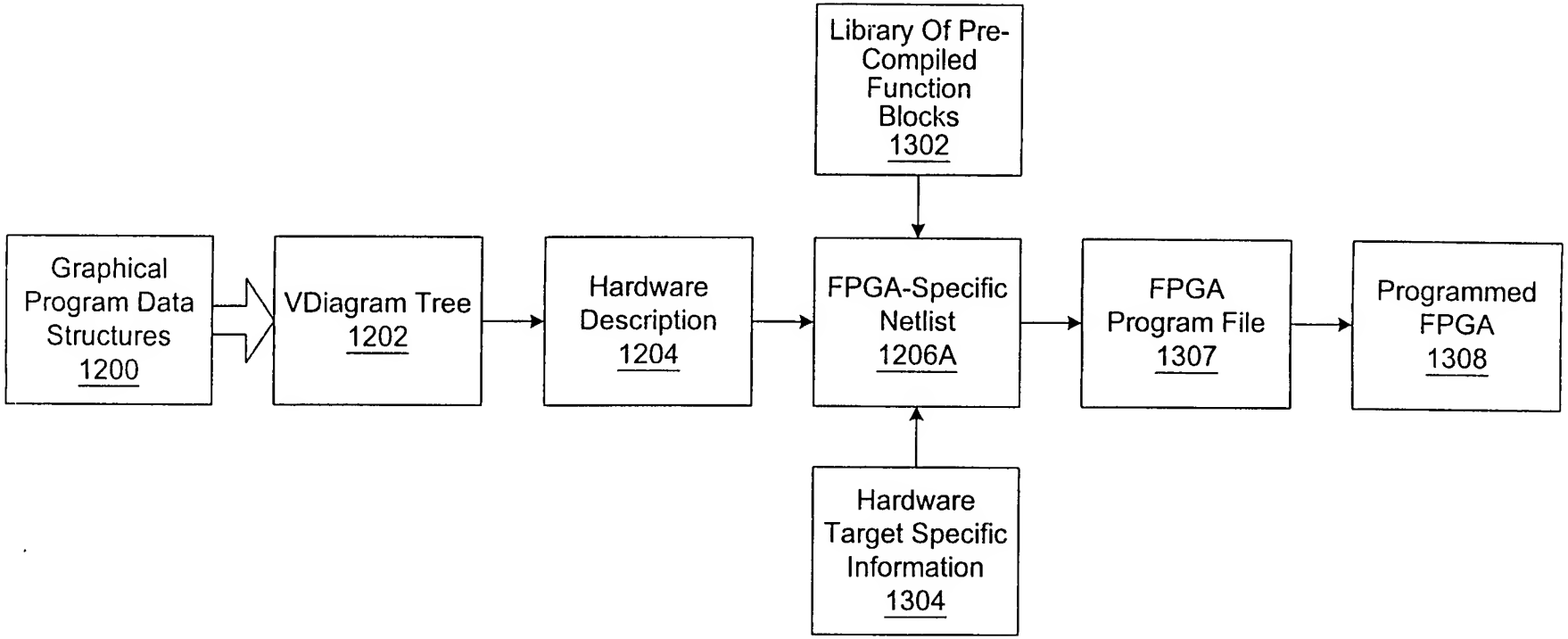
Library Of Pre-Compiled Function Blocks 1302
Graphical Program Data Structures 1200
VDiagram Tree 1202
Hardware Description 1204
FPGA-Specific Netlist 1206A
FPGA Program File 1307
Programmed FPGA 1308
Hardware Target Specific Information 1304

Figure 13

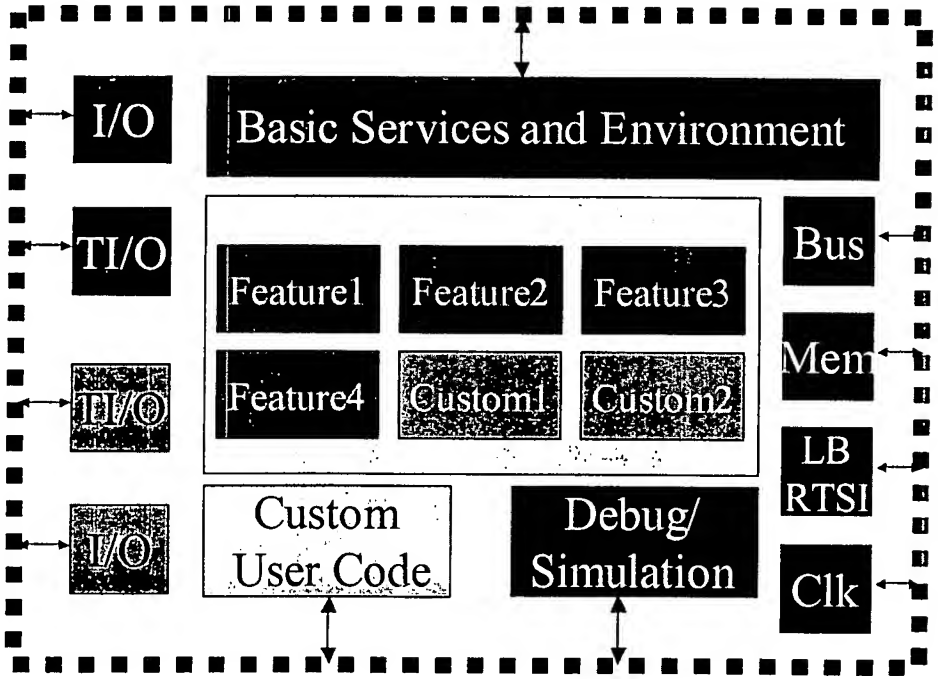
I/O
TI/O
TI/O
I/O
Basic Services and Environment
Feature1
Feature2
Feature3
Feature4
Custom1
Custom2
Custom User Code
Debug/ Simulation
Bus
Mem
LB RTSI
Clk

Figure 14

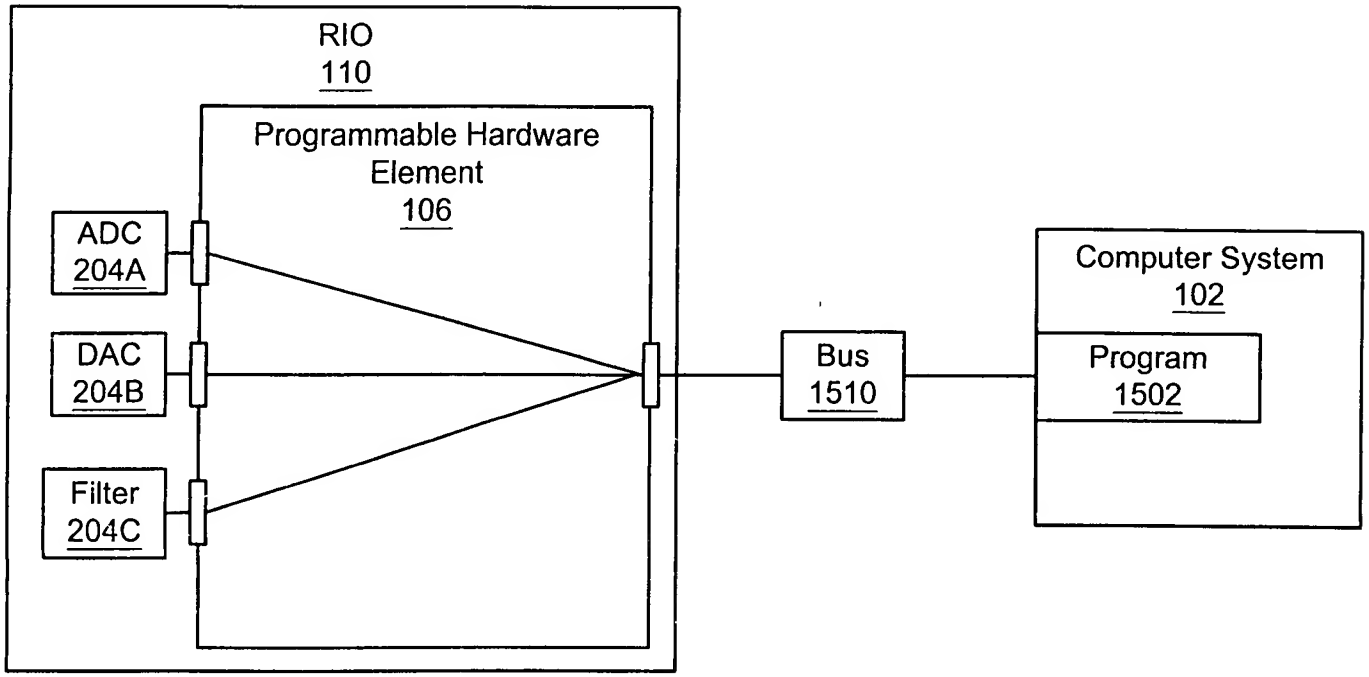
RIO
110
Programmable Hardware
Element
106
ADC
204A
DAC
204B
Filter
204C
Bus
1510
Computer System
102
Program
1502

Figure 15

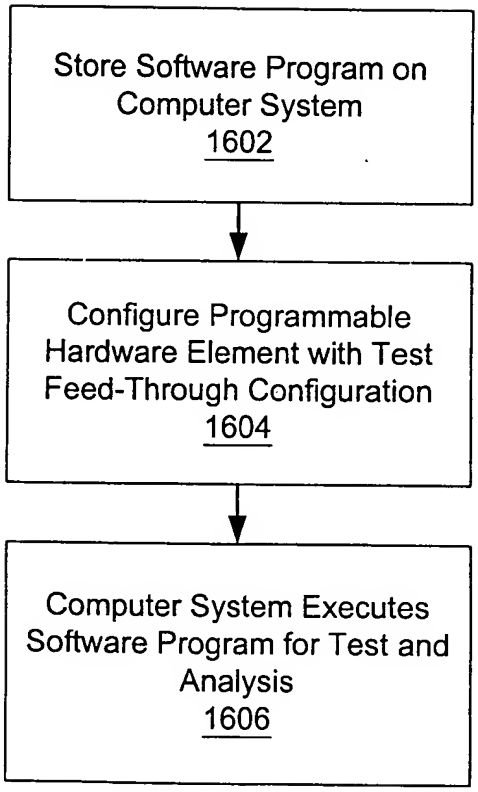
Store Software Program on Computer System
1602
Configure Programmable Hardware Element with Test Feed-Through Configuration
1604
Computer System Executes Software Program for Test and Analysis
1606

Figure 16

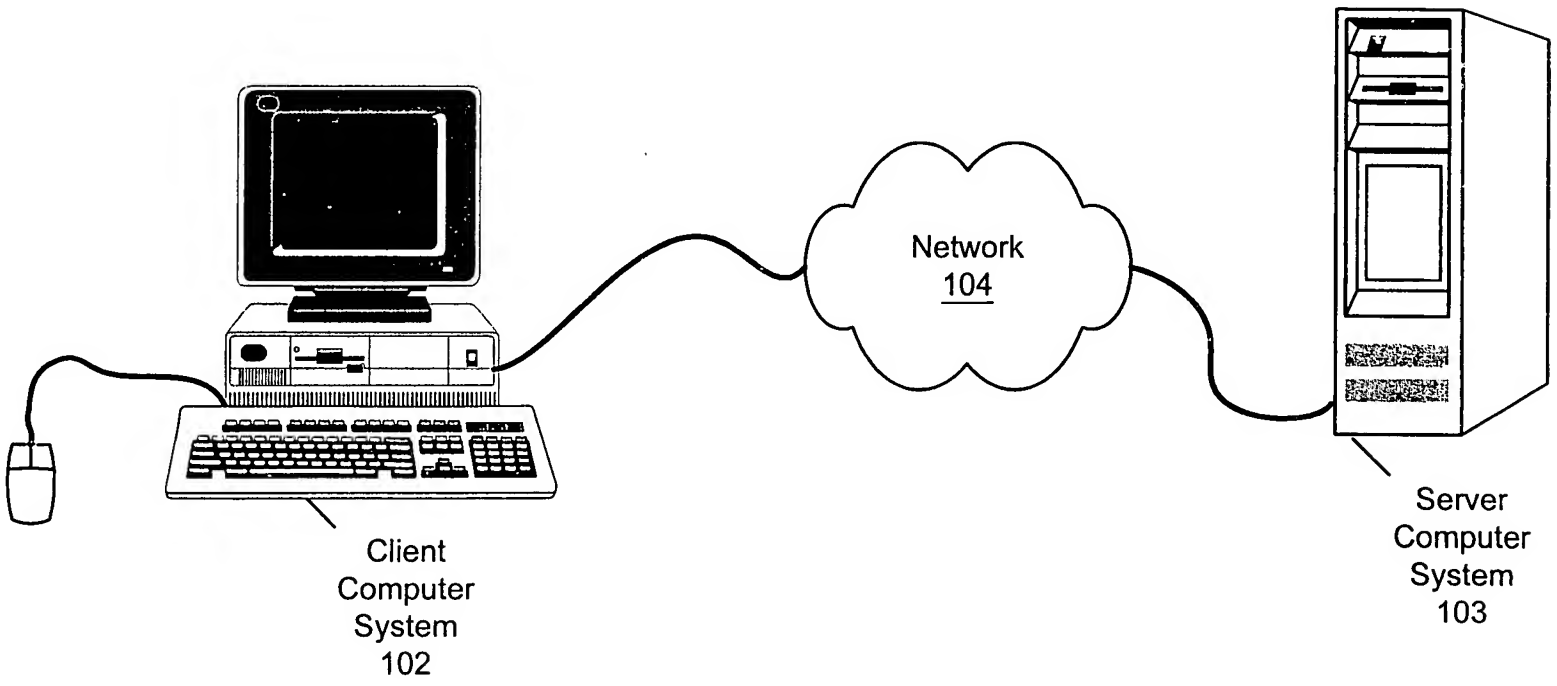
Network
104
Client
Computer
System
102
Server
Computer
System
103

Figure 17

4500504.140505

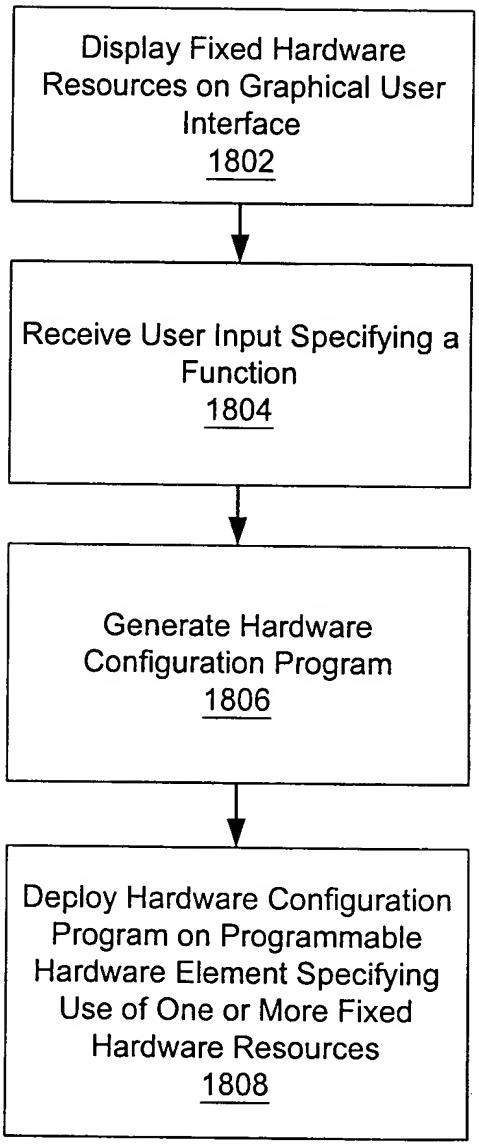

Figure 18

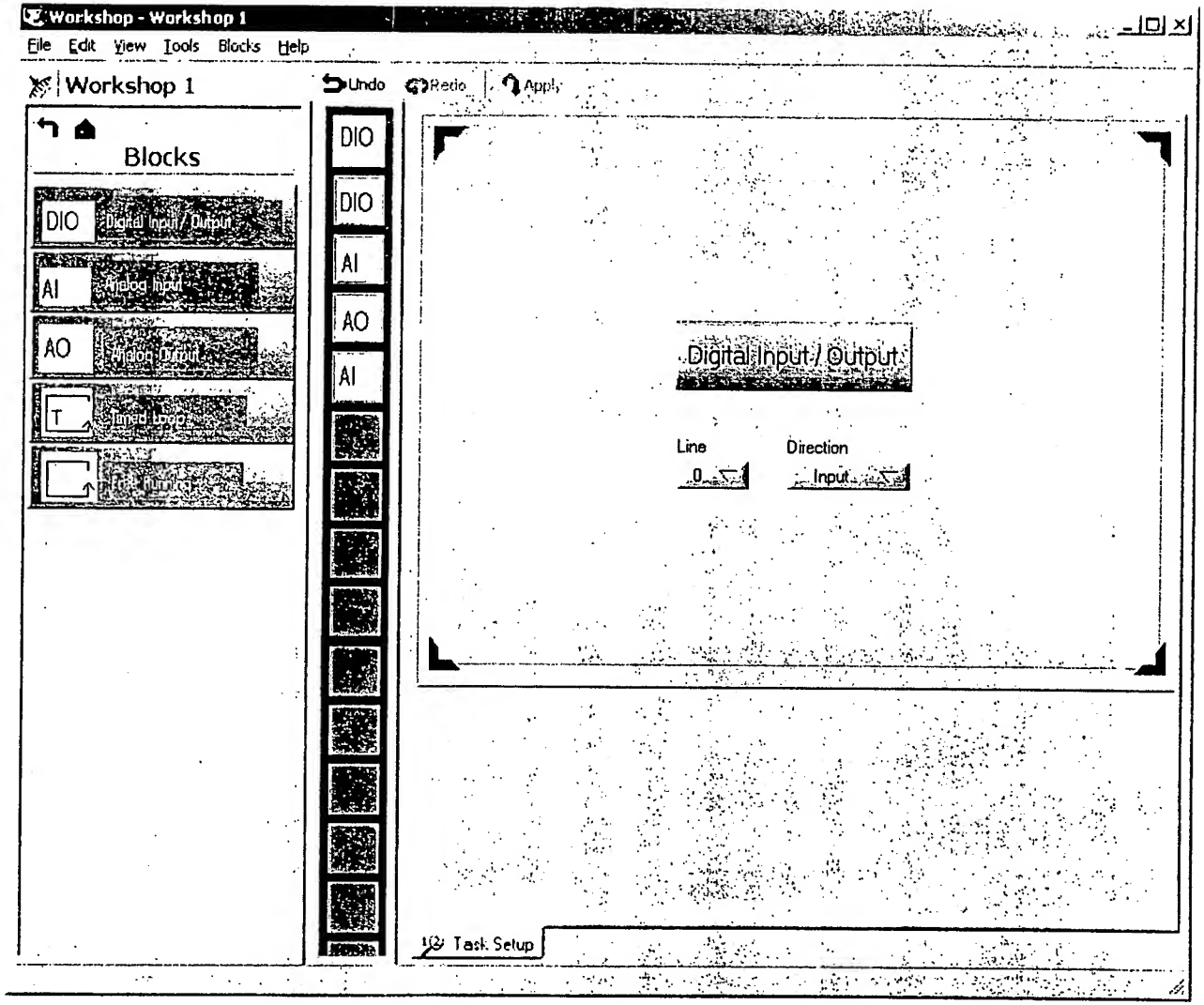
Workshop - Workshop 1
File Edit View Tools Blocks Help
Workshop 1
Undo
Redo
Apply
Blocks
DIO
Digital Input / Output
AI
Analog Input
AO
Analog Output
T
Timed Loop
DIO
DIO
AI
AO
AI
Digital Input / Output
Line
0
Direction
Input
Task Setup

Figure 19A

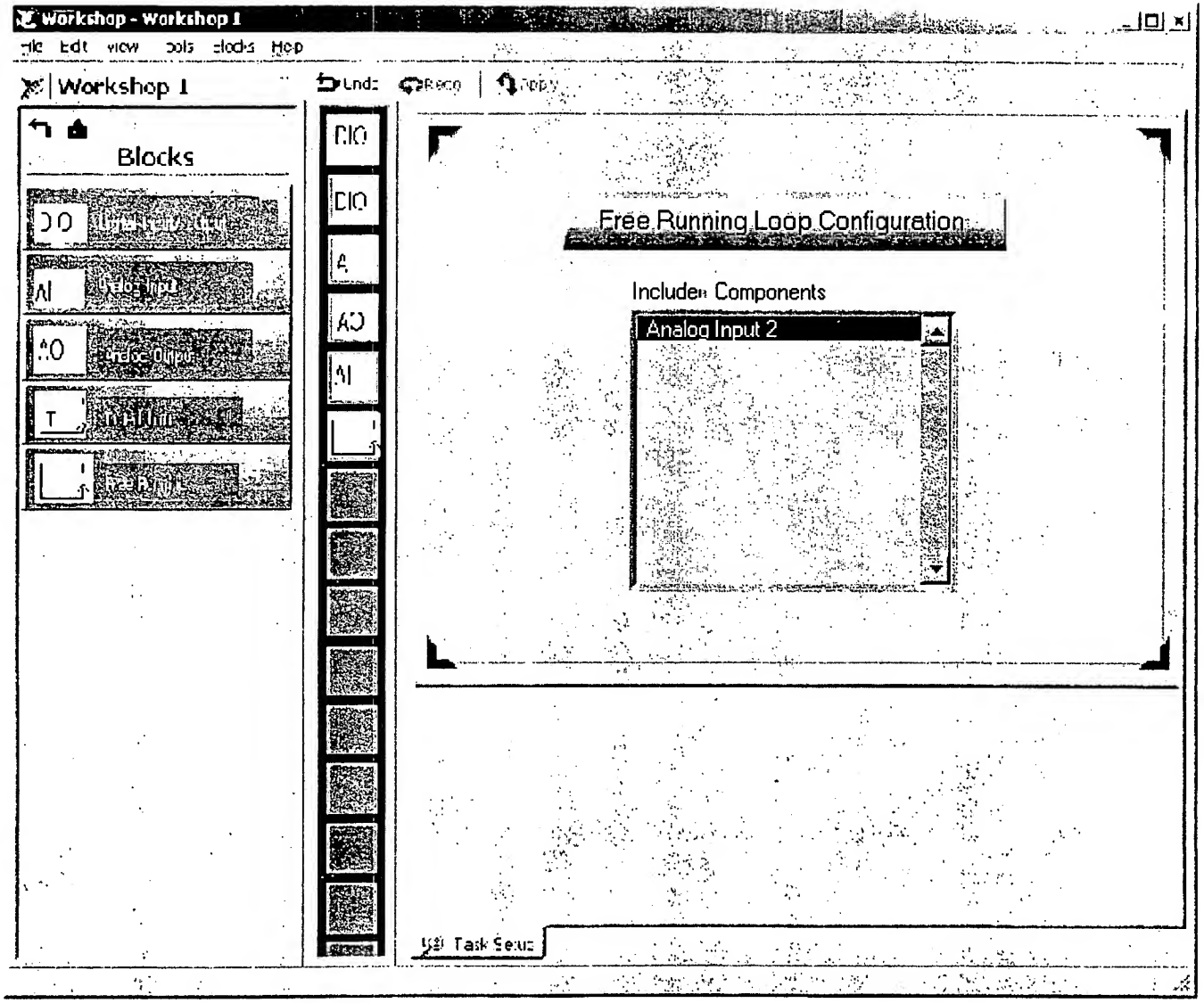
Workshop - Workshop 1
Workshop 1
Blocks
Free Running Loop Configuration
Included Components
Analog Input 2

Figure 19B

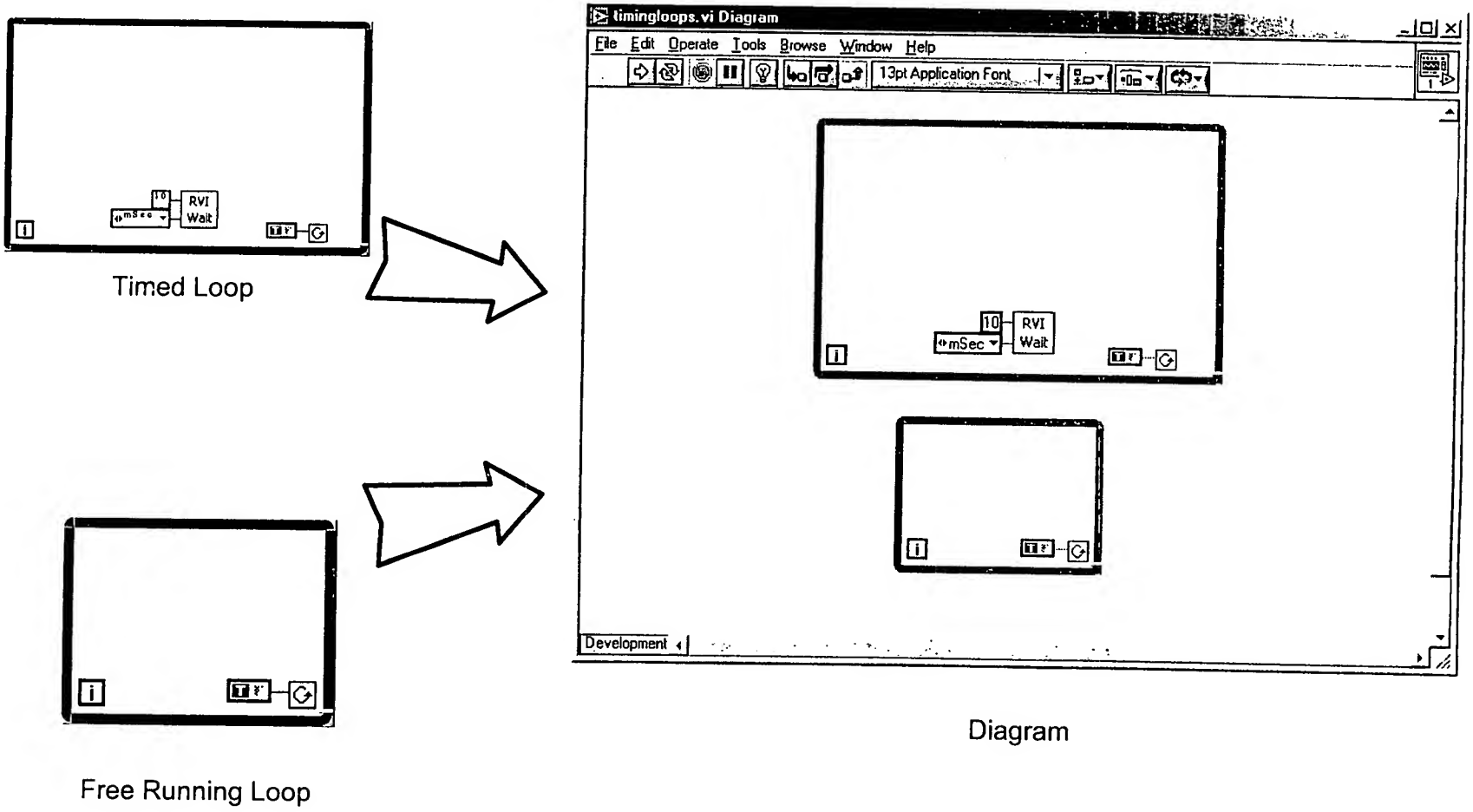
timingloops.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
10
mSec
RVI
Wait
Development
Timed Loop
Free Running Loop
Diagram

Figure 19C

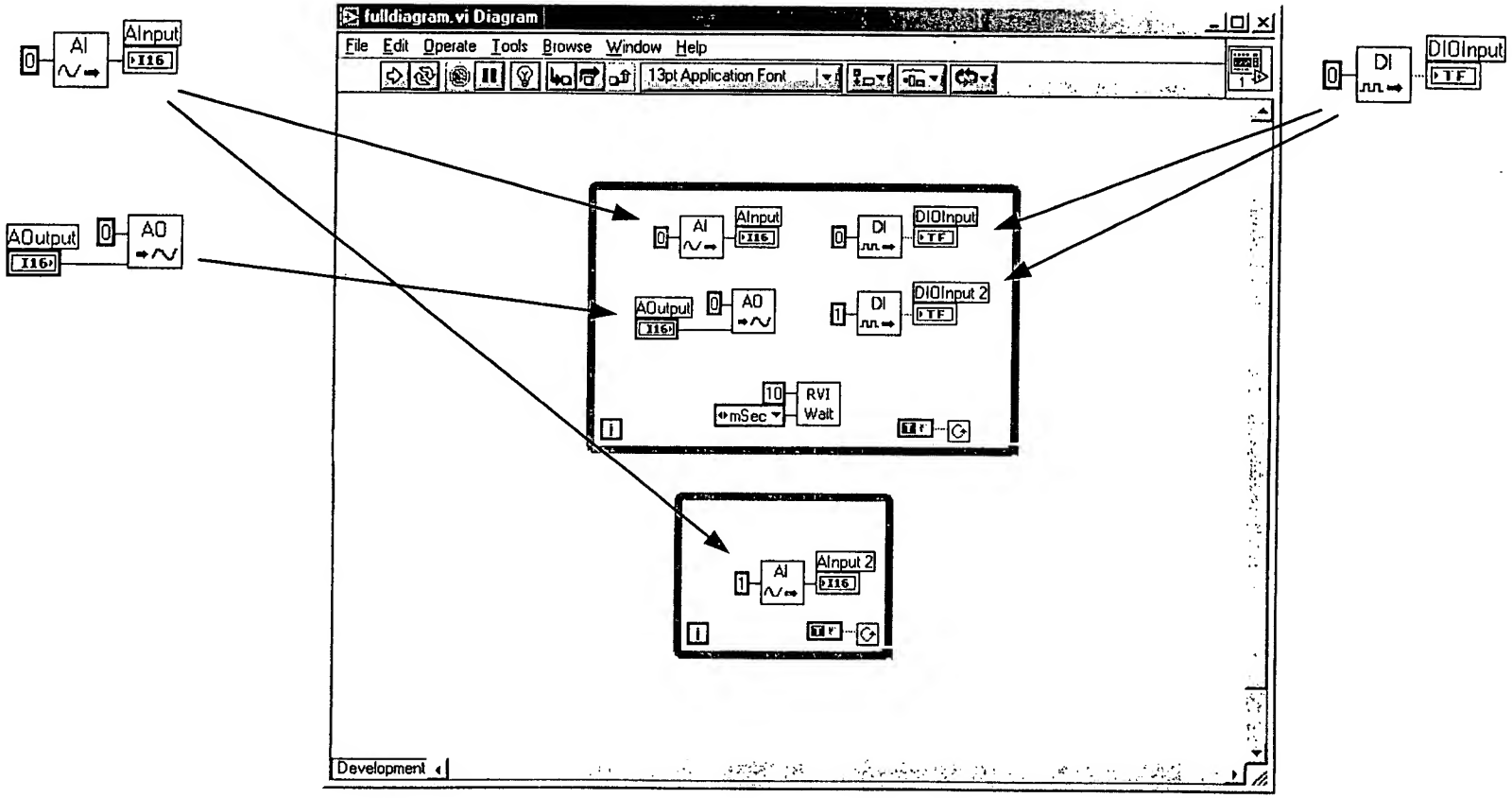
fulldiagram.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
AI
AInput
I16
AOutput
AO
DI
DIOInput
TF
DIOInput 2
10
mSec
RVI
Wait
AInput 2
Development

Figure 19D

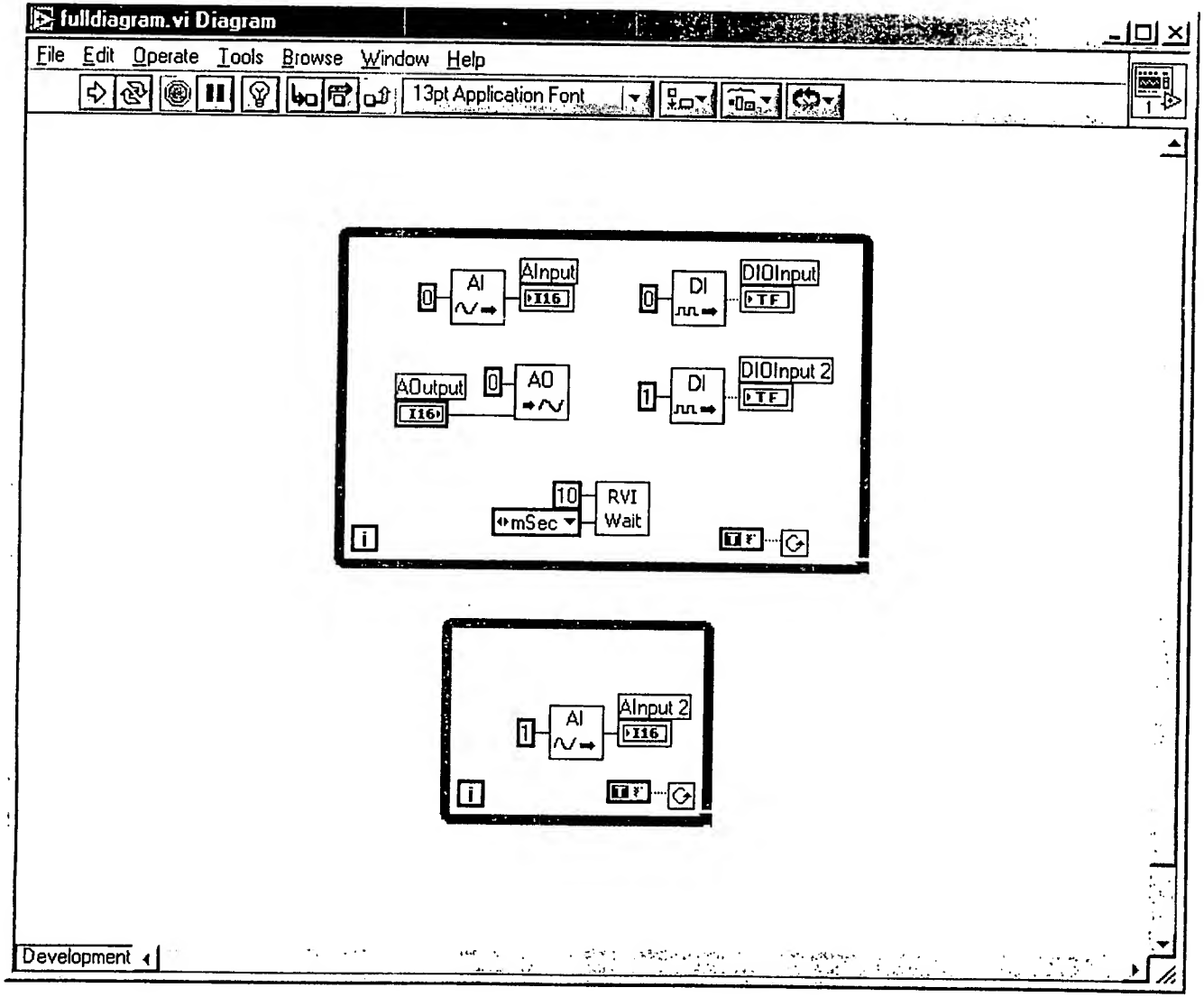
fulldiagram.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
AI
AInput
I16
DI
DIOInput
TF
AOutput
I16
AO
DI
DIOInput 2
TF
10
mSec
RVI
Wait
TF
AI
AInput 2
I16
TF
Development

Figure 19E

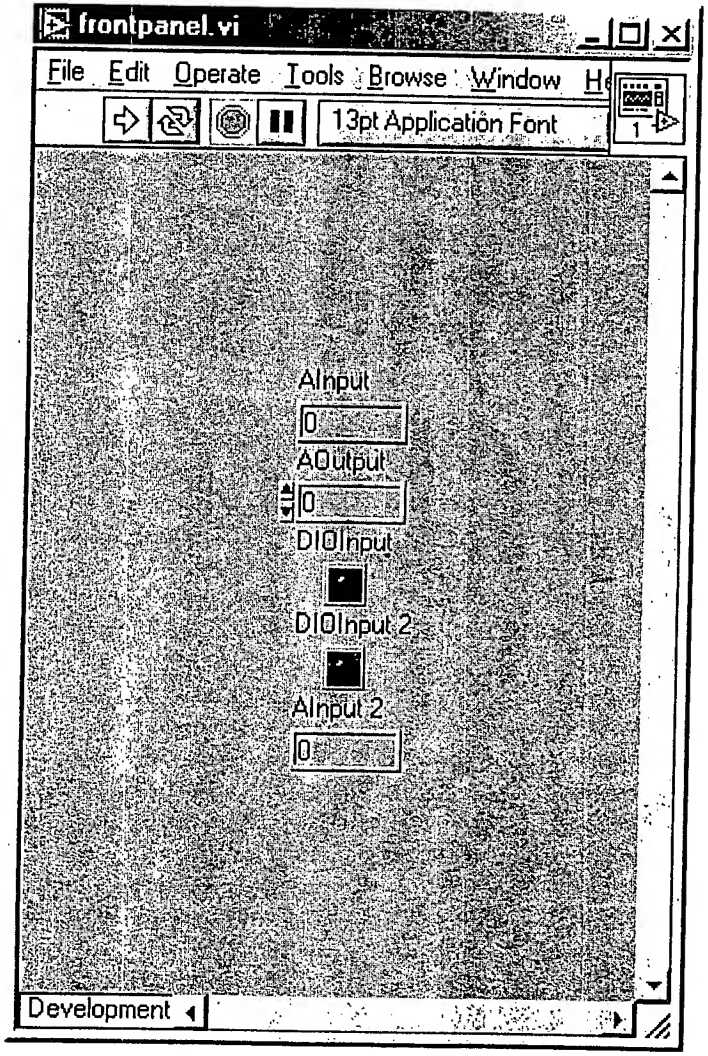
frontpanel.vi
File Edit Operate Tools Browse Window H
13pt Application Font
AInput
0
AOutput
0
DIOInput
DIOInput 2
AInput 2
0
Development

Figure 19F

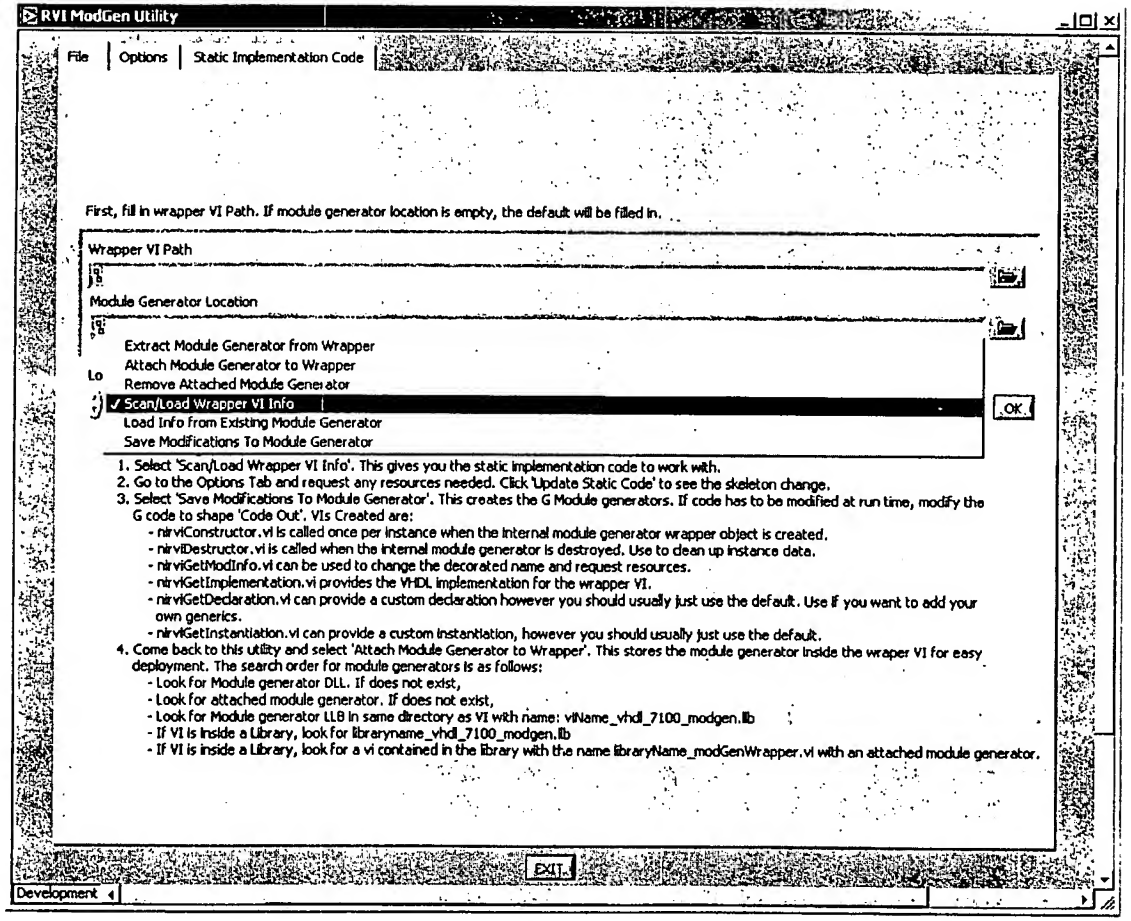
RVI ModGen Utility
File
Options
Static Implementation Code
First, fill in wrapper VI Path. If module generator location is empty, the default will be filled in.
Wrapper VI Path
Module Generator Location
Extract Module Generator from Wrapper
Attach Module Generator to Wrapper
Remove Attached Module Generator
✓ Scan/Load Wrapper VI Info
Load Info from Existing Module Generator
Save Modifications To Module Generator
OK
1. Select 'Scan/Load Wrapper VI Info'. This gives you the static implementation code to work with.
2. Go to the Options Tab and request any resources needed. Click 'Update Static Code' to see the skeleton change.
3. Select 'Save Modifications To Module Generator'. This creates the G Module generators. If code has to be modified at run time, modify the G code to shape 'Code Out'. VIs Created are:
- nirviConstructor.vi is called once per instance when the internal module generator wrapper object is created.
- nirviDestructor.vi is called when the internal module generator is destroyed. Use to clean up instance data.
- nirviGetModInfo.vi can be used to change the decorated name and request resources.
- nirviGetImplementation.vi provides the VHDL implementation for the wrapper VI.
- nirviGetDeclaration.vi can provide a custom declaration however you should usually just use the default. Use if you want to add your own generics.
- nirviGetInstantiation.vi can provide a custom instantiation, however you should usually just use the default.
4. Come back to this utility and select 'Attach Module Generator to Wrapper'. This stores the module generator inside the wraper VI for easy deployment. The search order for module generators is as follows:
- Look for Module generator DLL. If does not exist,
- Look for attached module generator. If does not exist,
- Look for Module generator LLB in same directory as VI with name: viName_vhdl_7100_modgen.llb
- If VI is inside a Library, look for libraryname_vhdl_7100_modgen.llb
- If VI is inside a Library, look for a vi contained in the library with the name libraryName_modGenWrapper.vi with an attached module generator.
EXIT
Development

Figure 20

4005504 v 4005504

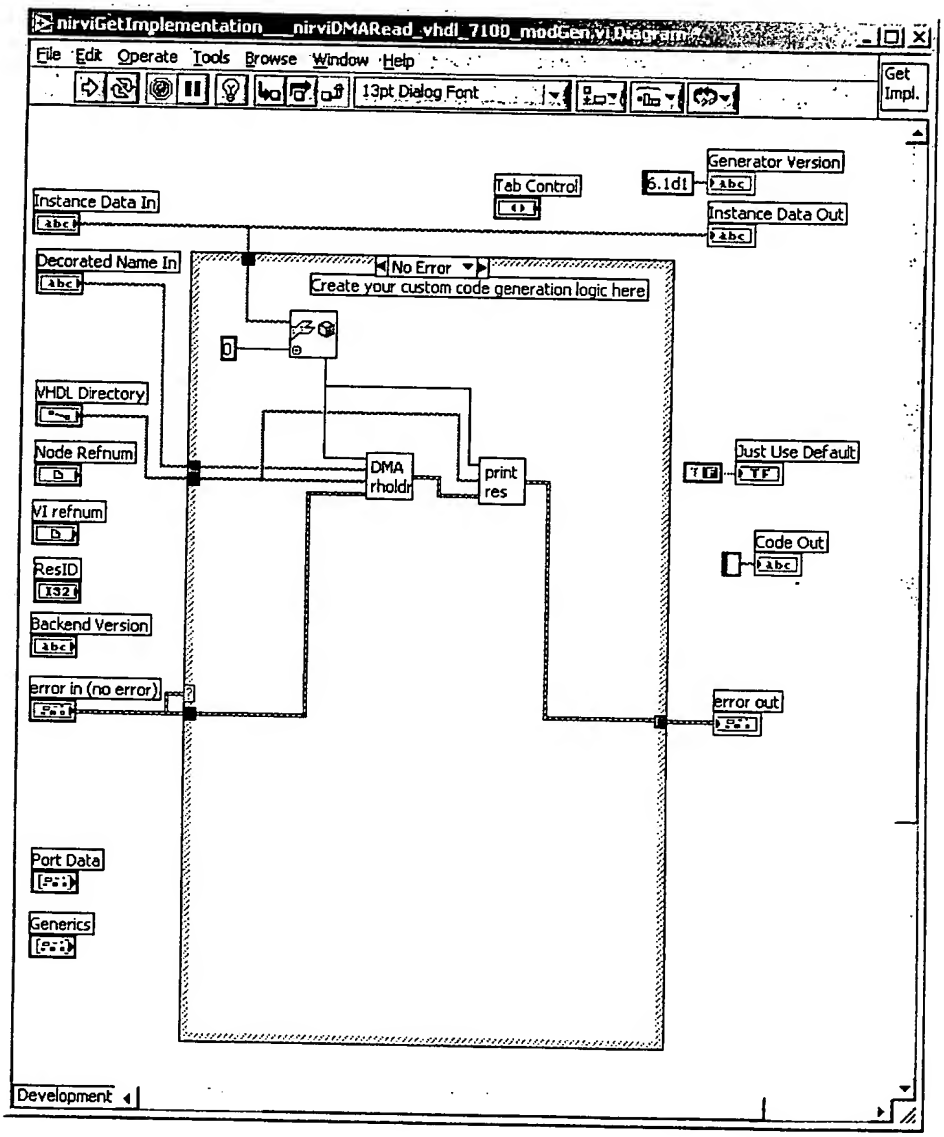

Figure 21

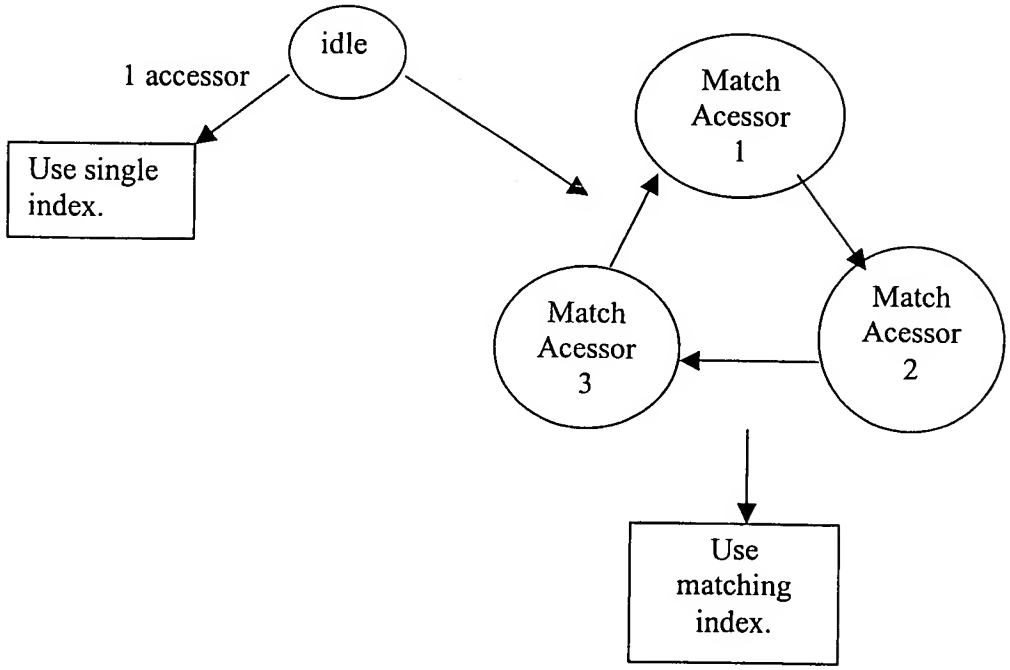
idle
1 accessor
Use single index.
Match Acessor 1
Match Acessor 2
Match Acessor 3
Use matching index.

Figure 22

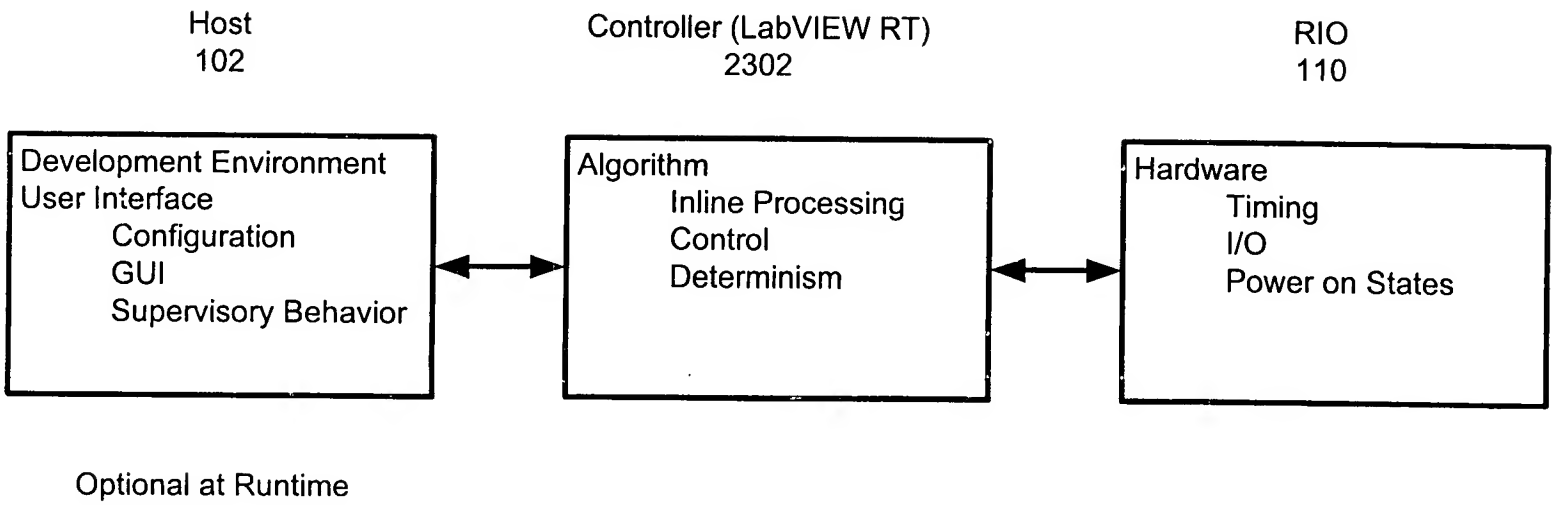
Host
102
Controller (LabVIEW RT)
2302
RIO
110
Development Environment
User Interface
Configuration
GUI
Supervisory Behavior
Algorithm
Inline Processing
Control
Determinism
Hardware
Timing
I/O
Power on States
Optional at Runtime

Figure 23A

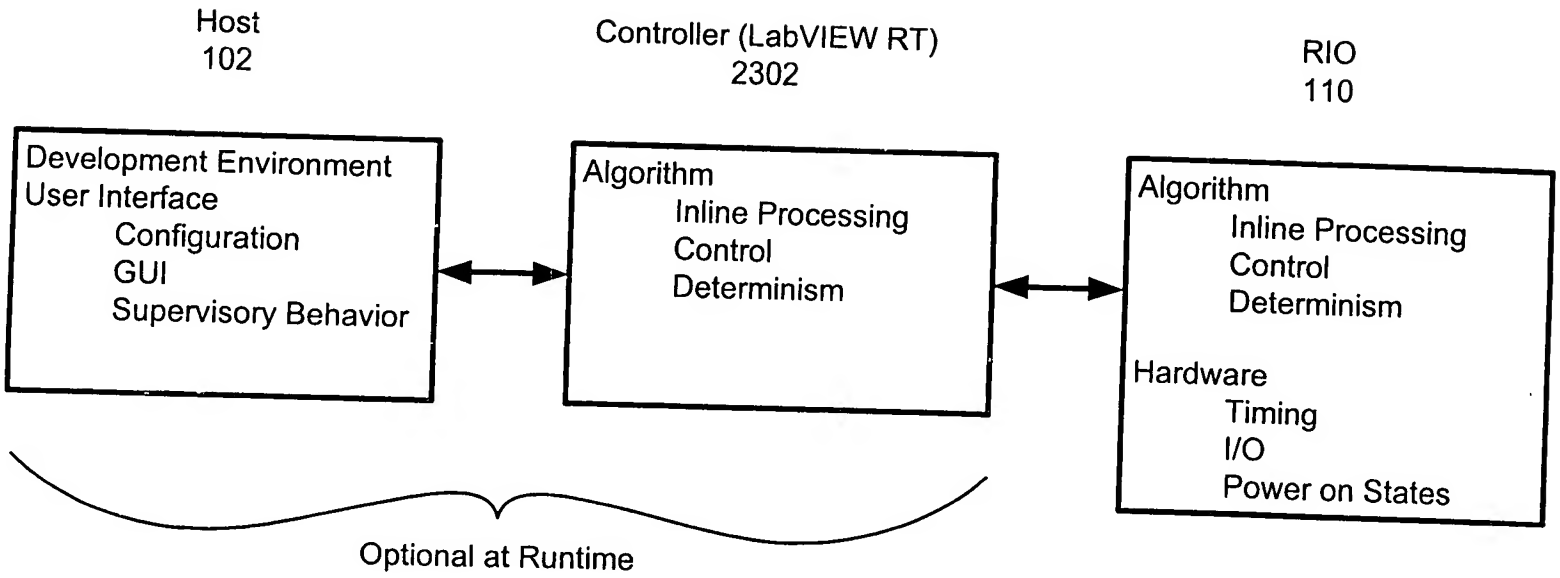
Host
102
Controller (LabVIEW RT)
2302
RIO
110
Development Environment
User Interface
Configuration
GUI
Supervisory Behavior
Algorithm
Inline Processing
Control
Determinism
Algorithm
Inline Processing
Control
Determinism
Hardware
Timing
I/O
Power on States
Optional at Runtime

Figure 23B

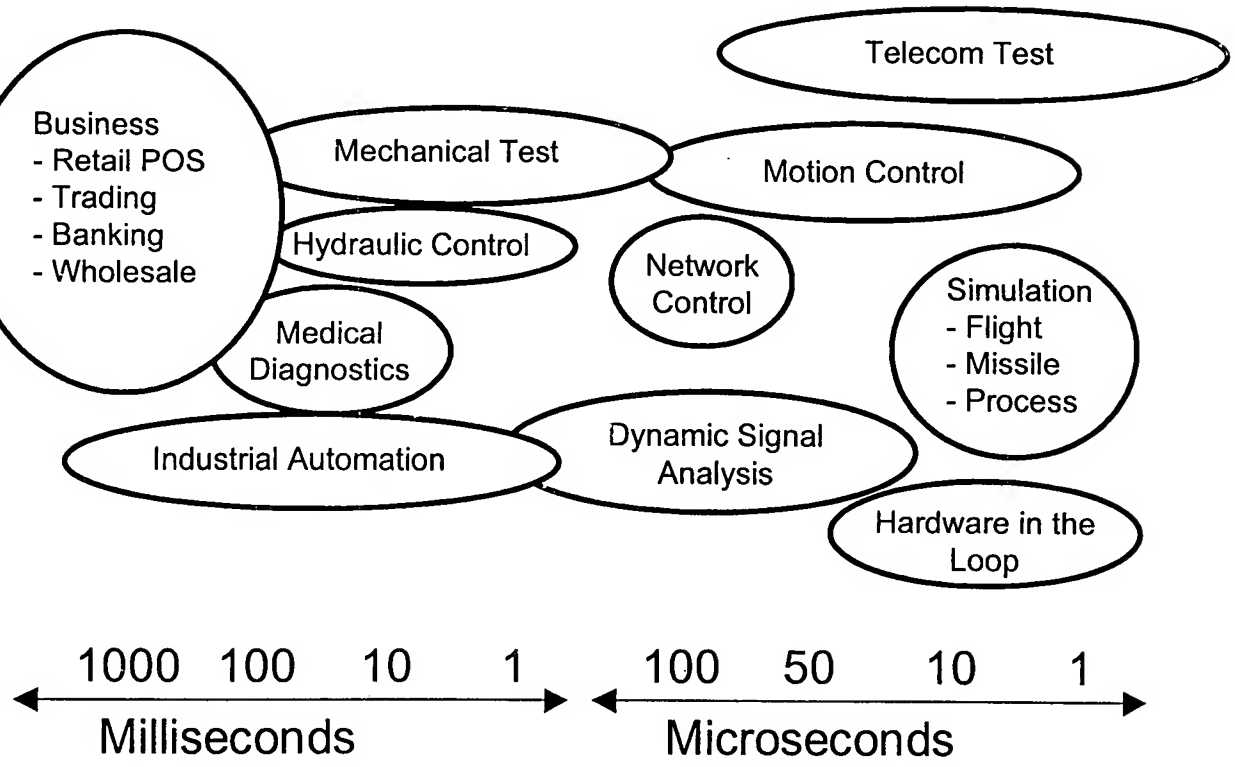
Telecom Test
Business
- Retail POS
- Trading
- Banking
- Wholesale
Mechanical Test
Motion Control
Hydraulic Control
Network Control
Simulation
- Flight
- Missile
- Process
Medical Diagnostics
Industrial Automation
Dynamic Signal Analysis
Hardware in the Loop
1000 100 10 1
Milliseconds
100 50 10 1
Microseconds

Figure 24